

Міністерство освіти і науки України
Луцький національний технічний університет

Лабораторний практикум з програмування мовою C/C++

*Навчальний посібник для студентів технічних спеціальностей
закладів вищої освіти I–IV рівнів акредитації*

Луцьк
Вежа-Друк
2020

УДК 004.43(075.8)
Л12

*Рекомендовано до друку
Луцьким національним технічним університетом
(протокол №3 від 3 грудня 2020 року)*

Рецензенти:

Пастернак Я. М. – доктор технічних наук, професор, завідувач кафедри прикладної математики та механіки Луцького НТУ;

Федонюк А. А. – кандидат фізико-математичних наук, доцент кафедри вищої математики та інформатики Східноєвропейського національного університету імені Лесі Українки;

Чухрій С. С. – директор ПП «Візор».

Автори-упорядники: Пех П. А., Лавренчук С. В., Делявський М. В., Гринюк С. В.

Лабораторний практикум з програмування мовою C/C++ : навч. посіб.
Л12 для студ. тех. спец. закл. вищ. освіти I–IV рівн. акредит. / П. А. Пех, С. В. Лавренчук, М. В. Делявський, С. В. Гринюк. – Луцьк : Вежа-Друк, 2020. – 228 с.

ISBN 978-966-940-335-3

Даний посібник-практикум – збірник програм мовою C/C++, виконання яких передбачається під час лабораторних занять з дисципліни «Програмування». Мета авторів – допомогти студентам в оволодінні ними практичними навичками у розробленні сучасних програмних продуктів, зокрема, в умовах дистанційної освіти. Окрім великої кількості програм, посібник також містить варіанти індивідуальних завдань.

Посібник передбачений для студентів технічних спеціальностей вищих навчальних закладів.

Усі назви програмних продуктів є зареєстрованими товарними марками відповідних фірм. Жодна частина цієї книги не може бути відтворена будь-якими засобами без дозволу видавництва.

УДК 004.43(075.8)

ISBN 978-966-940-335-3

© Пех П. А., Лавренчук С. В.,
Делявський М. В., Гринюк С. В. 2020
© Свиридюк К. А. (обкладинка), 2020

ЗМІСТ

ПЕРЕДМОВА	5
ЛАБОРАТОРНА РОБОТА №1: “ПРОГРАМУВАННЯ ЛІНІЙНИХ ПРОЦЕСІВ, ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ ЗАСОБАМИ МОВИ C”	6
ЛАБОРАТОРНА РОБОТА №2: “ПРОГРАМУВАННЯ ЛІНІЙНИХ ПРОЦЕСІВ, ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ ЗАСОБАМИ МОВИ C++”	17
ЛАБОРАТОРНА РОБОТА №3: “ФУНКЦІЇ КОРИСТУВАЧА З ПАРАМЕТРАМИ-ЗНАЧЕННЯМИ ЗАСОБАМИ МОВИ C/C++”	29
ЛАБОРАТОРНА РОБОТА №4: “ФУНКЦІЇ КОРИСТУВАЧА З ПАРАМЕТРАМИ-ВКАЗІВНИКАМИ ЗАСОБАМИ МОВИ C/C++”	31
ЛАБОРАТОРНА РОБОТА №5: “ПРОГРАМУВАННЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ ОПЕРАТОРА IF НА ПРИКЛАДІ ОБЧИСЛЕННЯ ЗНАЧЕННЯ СКЛАДНОЇ ФУНКЦІЇ”	33
ЛАБОРАТОРНА РОБОТА №6: “ПРОГРАМУВАННЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ ОПЕРАТОРА IF НА ПРИКЛАДІ ВИЗНАЧЕННЯ НАЛЕЖНОСТІ ТОЧКИ ОБЛАСТІ”	38
ЛАБОРАТОРНА РОБОТА №7: “ПРОГРАМУВАННЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ ОПЕРАТОРА ВИБОРУ SWITCH”	46
ЛАБОРАТОРНА РОБОТА №8: “ПРОГРАМУВАННЯ ЦИКЛІЧНИХ ПРОЦЕСІВ МОВОЮ C/C++ ЗА ДОПОМОГОЮ ОПЕРАТОРА FOR”	49
ЛАБОРАТОРНА РОБОТА №9: “ПРОГРАМУВАННЯ ЦИКЛІЧНИХ ПРОЦЕСІВ МОВОЮ C/C++ ЗА ДОПОМОГОЮ ОПЕРАТОРА WHILE”	56
ЛАБОРАТОРНА РОБОТА №10: “ПРОГРАМУВАННЯ ЦИКЛІЧНИХ ПРОЦЕСІВ МОВОЮ C/C++ ЗА ДОПОМОГОЮ ОПЕРАТОРА DO...WHILE”	62
ЛАБОРАТОРНА РОБОТА №11: “ПРОГРАМУВАННЯ ІТЕРАЦІЙНИХ ПРОЦЕСІВ МОВОЮ C/C++: ЧИСЕЛЬНЕ РОЗВ’ЯЗУВАННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ ПОЛОВИННОГО ДІЛЕННЯ.”	68
ЛАБОРАТОРНА РОБОТА №12: “ПРОГРАМУВАННЯ ІТЕРАЦІЙНИХ ПРОЦЕСІВ МОВОЮ C/C++: ЧИСЕЛЬНЕ РОЗВ’ЯЗУВАННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДАМИ ХОРД ТА ДОТИЧНИХ.”	74
ЛАБОРАТОРНА РОБОТА №13: “ОБЧИСЛЕННЯ З ЗАДАНОЮ ТОЧНІСТЮ СУМИ ЧЛЕНІВ ЗНАКОЗМІННОГО РЯДУ.”	83
ЛАБОРАТОРНА РОБОТА №14: “ОБЧИСЛЕННЯ З ЗАДАНОЮ ТОЧНІСТЮ СУМИ ЧЛЕНІВ ЗНАКОПОСТІЙНОГО РЯДУ”	87
ЛАБОРАТОРНА РОБОТА №15: “РЕКУРСИВНІ ФУНКЦІЇ МОВОЮ C/C++”	91
ЛАБОРАТОРНА РОБОТА №16: “РОБОТА З ОДНОВИМІРНИМИ МАСИВАМИ ЗАСОБАМИ МОВИ C/C++”	94
ЛАБОРАТОРНА РОБОТА №17: “ВПОРЯДКУВАННЯ ОДНОВИМІРНИХ МАСИВІВ ЗАСОБАМИ МОВИ C/C++”	105
ЛАБОРАТОРНА РОБОТА №18: “РОБОТА З ДВОВИМІРНИМИ МАСИВАМИ ЗАСОБАМИ МОВИ C/C++” ..	113
ЛАБОРАТОРНА РОБОТА №19: “ФОРМУВАННЯ ДВООВИМІРНИХ МАСИВІВ ЗА ЗАДАНИМ ЗАКОНОМ ЗАСОБАМИ МОВИ C/C++”	123
ЛАБОРАТОРНА РОБОТА №20: “РОБОТА З ФУНКЦІЯМИ ОБРОБКИ СИМВОЛІВ ТА СТРІЧОК ЗАСОБАМИ МОВИ C/C++”	130
ЛАБОРАТОРНА РОБОТА №21: “РОБОТА ЗІ СТРІЧКАМИ ЗАСОБАМИ МОВИ C/C++”	142

ЛАБОРАТОРНА РОБОТА №22: “РОБОТА ЗІ СТРУКТУРАМИ ТА ОБ’ЄДНАННЯМИ ЗАСОБАМИ МОВИ C/C++”	154
ЛАБОРАТОРНА РОБОТА №23: “РОБОТА З ФАЙЛАМИ ПОСЛІДОВНОГО ТА ПРЯМОГО ДОСТУПУ ЗАСОБАМИ МОВИ C”	165
ЛАБОРАТОРНА РОБОТА №24: “КОНСТРУЮВАННЯ КЛАСІВ ЗАСОБАМИ МОВИ C/C++”	172
ЛАБОРАТОРНА РОБОТА №25: “РОБОТА З КЛАСАМИ З РІЗНИМИ КОНСТРУКТОРАМИ ЗАСОБАМИ МОВИ C/C++”	187
ЛАБОРАТОРНА РОБОТА №26: “КОНСТРУЮВАННЯ БАЗОВИХ ТА ПОХІДНИХ КЛАСІВ ЗАСОБАМИ МОВИ C/C++”	198
ЛАБОРАТОРНА РОБОТА №27: “КОНСТРУЮВАННЯ КЛАСІВ З ВИКОРИСТАННЯМ ДРУЖНІХ ФУНКЦІЙ ЗАСОБАМИ МОВИ C/C++”	204
ЛАБОРАТОРНА РОБОТА №28: “РОБОТА З ОДНОЗВ’ЯЗНИМИ ЛІНІЙНИМИ СПИСКАМИ ЗАСОБАМИ МОВИ C/C++”	211
ЛАБОРАТОРНА РОБОТА №29: “РОБОТА З ДВОЗВ’ЯЗНИМИ ЛІНІЙНИМИ СПИСКАМИ ЗАСОБАМИ МОВИ C/C++”	216
ЛАБОРАТОРНА РОБОТА №30: “РОБОТА З ФАЙЛАМИ ЗАСОБАМИ МОВИ C++”	222
РЕКОМЕНДОВАНА ЛІТЕРАТУРА.....	226

ПЕРЕДМОВА

Метою лабораторного практикуму з програмування мовою C/C++ є допомога студентам у виконанні лабораторних робіт з дисципліни «Програмування» і оволодінні ними практичними навичками у розробленні сучасних програмних продуктів. Сформульована вище мета досягається за рахунок:

- поглибленого вивчення основ алгоритмізації інформаційних процесів;
- оволодіння сучасними технологіями програмування;
- вивчення принципів функціонування сучасних систем програмування, що дозволяють швидко і на сучасному рівні створювати прикладне програмне забезпечення;
- оволодіння практичними навичками розробки надійних і ефективних програм в середовищі Visual Studio та C++Builder.

Пристаючи до виконання лабораторної роботи з дисципліни «Програмування», студент повинен:

- опрацювати теоретичний матеріал з даної теми в обсязі, передбаченим навчальним планом для студентів даної спеціальності;
- володіти комп'ютером на достатньому рівні;
- мати навички роботи з операційною системою Windows та офісними додатками Word та Excel.

Даний посібник містить тридцять лабораторних робіт. Розроблені у них програми, як і програми, розроблені студентами самостійно повинні бути протестовані за різних наборів вхідних даних.

До програмних проектів, розроблюваних у процесі виконання індивідуальних завдань висувається ряд вимог, яких студент має дотримуватися. Програмні проекти повинні:

- бути виконані у відповідності до умов завдання;
- містити самостійно розроблені та протестовані програми;
- ґрунтуватися на результатах самостійної роботи (дослідженнях);
- бути оформленими за державними стандартами.

ЛАБОРАТОРНА РОБОТА №1: “ПРОГРАМУВАННЯ ЛІНІЙНИХ ПРОЦЕСІВ, ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ ЗАСОБАМИ МОВИ C”

МЕТА РОБОТИ

1. Вивчити структуру програм мовою C.
2. Вивчити питання алгоритмізації типових обчислювальних процесів.
3. Ознайомитися з етапами виконання програм засобами мови C.
4. Навчитися складати прості програми лінійних процесів засобами мови C.
5. Вивчити формати введення та виведення даних різних типів засобами мови C.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати лінійну програму і зберегти її під назвою `lab_work_01_01` для розв’язування наступної задачі. Дано: R - радіус основи прямого кругового циліндра; h - висота прямого кругового циліндра. Необхідно обчислити та вивести на друк: $sb=2\pi r h$ - площу бічної поверхні циліндра; $sp=sb+2\pi r^2$ - площу повної поверхні циліндра; $v=\pi r^2 h$ - об'єм циліндра. Результати розрахунків подати у вигляді таблиці. Завдання виконати мовою C.

```
/*
lab_work_01_01.
Лінійні обчислювальні процеси засобами мови C.
Дано: R- радіус основи прямого кругового
циліндра; h - висота прямого кругового циліндра.
Обчислити та вивести на друк:
sb=2*pi*r*h - площу бічної поверхні циліндра;
sp=sb+2*pi*r*r - площу повної поверхні циліндра;
v=pi*r*r*h - об'єм циліндра.
Результати розрахунків подати у вигляді таблиці.
*/

#include <stdio.h>
#include <conio.h>
#include <math.h>

int main() {
    clrscr(); // очищення екрану

    // Оголошення констант та змінних:

    const float pi=3.14159; // оголошення константи pi
    float r; // радіус основи прямого кругового циліндра
    float h; // висота прямого кругового циліндра
    float sb; // площа бічної поверхні циліндра
    float sp; // площа повної поверхні циліндра
    float v; // об'єм циліндра

    // Ввід заданих величин:
    printf (" Введіть радіус основи циліндра: ");
    scanf ("%f",&r);
```

```

printf ("   Введіть висоту циліндра: ");
scanf ("   %f",&h);

// Обчислення параметрів циліндра:
sb=2*pi*r*h;
sp=sb+2*pi*r*r*h;
v=pi*r*h*h;

// Друк результатів у вигляді таблиці:
printf("\n   Вхідні   величини та результати обчислень:");
printf("\n   I-----I");
printf("\n   I           Вхідні   величини:           I");
printf("\n   I-----I");
printf("\n   I   Радіус основи циліндра   I   %8.2f   I",r);
printf("\n   I-----I");
printf("\n   I   Висота циліндра           I   %8.2f   I",h);
printf("\n   I-----I");
printf("\n   I           Результати обчислень:           I");
printf("\n   I-----I");
printf("\n   I   Бічна поверхня циліндра   I   %8.2f   I",sb);
printf("\n   I-----I");
printf("\n   I   Повна поверхня циліндра   I   %8.2f   I",sp);
printf("\n   I-----I");
printf("\n   I   Об'єм циліндра           I   %8.2f   I",v);
printf("\n   I-----I\n");

getch();      //Затримка результатів обчислення на екрані
return 0;
}

```

Результати тестування програми:

Введіть радіус основи циліндра: 4
Введіть висоту циліндра: 5

```

Вхідні   величини та результати обчислень:
I-----I
I           Вхідні   величини:           I
I-----I
I   Радіус основи циліндра   I   4           I
I-----I
I   Висота циліндра           I   5           I
I-----I
I           Результати обчислень:           I
I-----I
I   Бічна поверхня циліндра   I   125.664      I
I-----I
I   Повна поверхня циліндра   I   628.318      I
I-----I
I   Об'єм циліндра           I   314.159      I
I-----I

```

2. Ввести, відлагодити та протестувати програму форматowanego виведення даних базових типів мови C і зберегти її під назвою *lab_work_01_02*.

```
/* **** */
/* lab_work_01_02. */
/* Ввести, відлагодити та протестувати програму */
/* засобами мови C, у якій передбачити виведення на */
/* екран даних цілого, дійсного та символьного типів */
/* з використанням відповідних форматів. */
/* **** */
#include <stdio.h>
#include <conio.h>

int main()
{clrscr();
  char symv = 'A';

  // Форматований вивід даних цілих типів
  printf(" Друк десяткового числа 455 форматом %%d :\n");
  printf(" %d\n",455);
  printf(" Друк десяткового числа 455 форматом %%i :\n");
  printf(" %i\n",455);
  printf(" Друк десяткового числа +455 форматом %%d :\n");
  printf(" %d\n",+455);
  printf(" Друк десяткового числа -455 форматом %%d :\n");
  printf(" %d\n",-455);
  printf(" Друк десяткового числа 32000 форматом %%hd :\n");
  printf(" %hd\n",32000);
  printf(" Друк числа 2000000000 форматом %%ld :\n");
  printf(" %ld\n",2000000000);
  printf(" Друк вісімкового числа 455 форматом %%o :\n");
  printf(" %o\n",455);
  printf(" Друк числа -455 форматом %%u :\n");
  printf(" %u\n",-455);
  printf(" Друк шістнадцяткового числа 455 форматом %%x :\n");
  printf(" %x\n",455);
  printf(" Друк шістнадцяткового числа 455 форматом %%X :\n");
  printf(" %X\n\n",455);

  // Форматований вивід даних дійсних типів
  printf(" Друк дійсного числа 1234567.89 форматом %%e :\n");
  printf(" %e\n",1234567.89);
  printf(" Друк дійсного числа +1234567.89 форматом %%e :\n");
  printf(" %e\n",+1234567.89);
  printf(" Друк дійсного числа -1234567.89 форматом %%e :\n");
  printf(" %e\n",-1234567.89);
  printf(" Друк дійсного числа -1234567.89 форматом %%e :\n");
  printf(" %e\n",-1234567.89);
  printf(" Друк дійсного числа 1234567.89 форматом %%E :\n");
  printf(" %E\n",1234567.89);
  printf(" Друк дійсного числа 1234567.89 форматом %%f :\n");
  printf(" %f\n",1234567.89);
  printf(" Друк дійсного числа 1234567.89 форматом %%g :\n");
```



```

printf("  %g\n",1234567.89);
printf("  Друк дійсного числа 1234567.89  форматом %%G :\n");
printf("  %G\n\n",1234567.89);

// Форматований вивід даних символного типу
char str[] = "Це стрічка";
const char *strPtr="Це також стрічка";
printf("\n  Друк символу 'A' за форматом %%c:");
printf("\n  %c",symv);
printf("\n  Друк стрічки-константи за форматом %%s:");
printf("\n  %s","Це стрічка");
printf("\n  Друк стрічки з ім'ям str за форматом %%s:");
printf("\n  %s",str);
printf("\n  Друк стрічки з допомогою вказівника strPtr за
        форматом %%s:");
printf("\n  %s",strPtr);

getch();
return 0;
}

```

Результати тестування програми:

```

Друк десяткового числа 455 форматом %d :
455
Друк десяткового числа 455 форматом %i :
455
Друк десяткового числа +455 форматом %d :
455
Друк десяткового числа -455 форматом %d :
-455
Друк десяткового числа 32000 форматом %hd :
32000
Друк числа 1234567890 форматом %ld :
1234567890
Друк вісімкового числа 455 форматом %o :
707
Друк числа -455 форматом %u :
65081
Друк шістнадцяткового числа 455 форматом %x :
1c7
Друк шістнадцяткового числа 455 форматом %X :
1C7

Друк дійсного числа 1234567.89  форматом %e :
1.234568e+06
Друк дійсного числа +1234567.89  форматом %e :
1.234568e+06
Друк дійсного числа -1234567.89  форматом %e :
-1.234568e+06
Друк дійсного числа -1234567.89  форматом %e :
-1.234568e+06
Друк дійсного числа 1234567.89  форматом %E :
1.234568E+06

```

Друк дійсного числа 1234567.89 форматом %f :
1234567.890000
Друк дійсного числа 1234567.89 форматом %g :
1.23457e+06
Друк дійсного числа 1234567.89 форматом %G :
1.23457E+06

Друк символу 'A' за форматом %c:
A
Друк стрічки-константи за форматом %s:
Це стрічка
Друк стрічки з ім'ям str за форматом %s:
Це стрічка
Друк стрічки з допомогою вказівника strPtr за форматом %s:
Це також стрічка

3. Ввести, відлагодити та протестувати програму форматowanego виведення даних базових типів даних мови C з використанням форматів d, f, e, E, g, G, r та прапорців '-' '+' ' ' ' ' '#' '0' і зберегти її під назвою *lab_work_01_03*.

```
/* **** */
/* lab_work_01_03 */
/* Виведення даних різних типів з використанням */
/* форматів d, f, e, E, g, G, r та прапорців '-' '+' ' ' ' ' '#' '0' */
/* для форматowanego виводу даних засобами мови C. */
/* **** */

#include <stdio.h>
#include <conio.h>

int main()
{clrscr();

    // Друк додатніх цілих чисел
    // з одночасним вирівнюванням по правому краю:
    printf("\n%s\n%s\n",
        " Друк додатніх цілих чисел за форматом %4d",
        " з одночасним вирівнюванням по правому краю:");
    printf(" %4d\n",1);
    printf(" %4d\n",12);
    printf(" %4d\n",123);
    printf(" %4d\n",1234);
    printf(" %4d\n",12345);
    printf("%s\n%s\n",
        " В останньому випадку бракувало ширини формату-",
        " цим пояснюється неякісне вирівнювання");

    printf("\n%s\n%s\n",
        " Друк від'ємних цілих чисел за форматом %4d",
        " з одночасним вирівнюванням по правому краю:");
    printf(" %4d\n",-1);
    printf(" %4d\n",-12);
```

```

printf(" %4d\n",-123);
printf(" %4d\n",-1234);
printf(" %4d\n",-12345);
printf("%s\n%s\n",
    " В останніх двох випадках бракувало ширини формату -",
    " цим пояснюється неякісне вирівнювання");
getch();

clrscr();
// Використання у форматах прапорця (-)
// для вирівнювання по лівому краю:
printf("\n%s\n%s\n",
    " Друк констант різного типу за форматами з восьми",
    " символів",
    " з одночасним вирівнюванням по правому краю:");
printf(" %8s%8d%8c%8f\n", "Привіт",7,'a',1.23);
printf("\n%s\n%s\n",
    " Використання у форматах прапорця (-) ",
    " для вирівнювання по лівому краю:");
printf(" %-8s%-8d%-8c%-8f\n", "Привіт",7,'a',1.23);
getch();

clrscr();
// Використання у форматах прапорця (+)
// для обов'язкового друку знака чисел:
printf("\n%s\n%s\n",
    " Друк констант цілого типу за форматом %d ",
    " з одночасним вирівнюванням по правому краю:");
printf(" %d\n %d\n",786,-786);
printf("\n%s\n%s\n",
    " Використання у форматах прапорця (+) ",
    " для обов'язкового друку знака чисел:");
printf(" %+d\n %+d\n",786,-786);
getch();

clrscr();
// Використання у форматах прапорця (пробіл)
// для друку знака пробіл перед додатніми числами:
printf("\n%s\n%s\n",
    " Друк констант цілого типу за форматом %d ",
    " з одночасним вирівнюванням по правому краю:");
printf(" %d\n %d\n",786,-786);
printf("\n%s\n%s\n",
    " Використання у форматах прапорця (пробіл) ",
    " для друку знака пробіл перед додатніми числами:");
printf(" %d\n %d\n",786,-786);
getch();

clrscr();
// Використання у форматах прапорця (#) для друку:
// знака o - перед вісімковими числами
// знака x - перед шістнадцятковими числами
// крапки - в дійсних числах

```

```

int c=1427;
double p=1427.0;
printf("\n%s\n%s\n%s\n%s\n%s\n%s\n",
    " Використання у форматах прапорця (#) ",
    " для друку:",
    " знака o - перед вісімковими числами",
    " знака x - перед шістнадцятковими числами",
    " знака X - перед шістнадцятковими числами",
    " крапки - в дійсних числах");
printf("\n  %#o",c);
printf("\n  %#x",c);
printf("\n  %#X",c);
printf("\n  %g",p);
printf("\n  %#g\n",p);
getch();

clrscr();
// Використання у форматах прапорця (0)
// для заповнення порожніх символів формату нулями:
printf("\n%s\n%s\n",
    " Використання у форматах прапорця (0) ",
    " для заповнення порожніх символів формату нулями:");
printf("  %+09d\n",452);
printf("  %09d\n",452);
getch();

return 0;
}

```

Результати тестування програми:

Друк додатніх цілих чисел за форматом %4d
з одночасним вирівнюванням по правому краю:

```

1
12
123
1234
12345

```

В останньому випадку бракувало ширини формату -
цим пояснюється неякісне вирівнювання

Друк від'ємних цілих чисел за форматом %4d
з одночасним вирівнюванням по правому краю:

```

-1
-12
-123
-1234
-12345

```

В останніх двох випадках бракувало ширини формату -
цим пояснюється неякісне вирівнювання

Друк констант різного типу за форматами з восьми символів
з одночасним вирівнюванням по правому краю:

Привіт 7 a1.230000

Використання у форматах прапорця (-)
для вирівнювання по лівому краю:
Привіт 7 а 1.230000

Друк констант цілого типу за форматом %d
з одночасним вирівнюванням по правому краю:
786
-786

Використання у форматах прапорця (+)
для обов'язкового друку знака чисел:
+786
-786

Друк констант цілого типу за форматом %d
з одночасним вирівнюванням по правому краю:
786
-786

Використання у форматах прапорця (пробіл)
для друку знака пробіл перед додатніми числами:
786
-786

Використання у форматах прапорця (#)
для друку:
знака o - перед вісімковими числами
знака x - перед шістнадцятковими числами
знака X - перед шістнадцятковими числами
крапки - в дійсних числах

02623
0x593
0X593
1427
1427.00

Використання у форматах прапорця (0)
для заповнення порожніх символів формату нулями:
+00000452
000000452

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати лінійну програму розв'язування задачі , вибраної згідно з варіантом.

1. Дано сторони a , b , c трикутника. Обчислити висоти трикутника:

$$h_a = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{a};$$

$$h_b = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{b};$$

$$h_c = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{ac}.$$

де $p = \frac{a+b+c}{2}$ – півпериметр.

2. Дано сторони a , b , c трикутника. Обчислити медіани трикутника:

$$m_a = \frac{\sqrt{2b^2 + 2c^2 - a^2}}{2};$$

$$m_b = \frac{\sqrt{2b^2 + 2c^2 - a^2}}{2};$$

$$m_c = \frac{\sqrt{2b^2 + 2c^2 - a^2}}{2};$$

3. Дано катети a , b прямокутного трикутника. Обчислити гіпотенузу $c = \sqrt{a^2 + b^2}$ та площу $S = \frac{ab}{2}$ трикутника:

4. Дано сторони a , b , c трикутника. Знайти площу цього трикутника за формулою Герона $S = \sqrt{p(p-a)(p-b)(p-c)}$, де $p = \frac{a+b+c}{2}$ – півпериметр.

5. Дано основи трапеції a , b та її висота h . Обчислити середню лінію трапеції $s = \frac{a+b}{2}$ та її площу $S = \frac{a+b}{2} h$.

6. Дано радіус кола R . Обчислити довжину кола $l = 2\pi R$ та площу круга $S = \pi R^2$.

7. Дано сторони a , b , c трикутника. Обчислити радіус описаного кола $R = \frac{abc}{4\sqrt{p(p-a)(p-b)(p-c)}}$ та радіус вписаного кола $r = \frac{\sqrt{p(p-a)(p-b)(p-c)}}{p}$, де $p = \frac{a+b+c}{2}$ – півпериметр.

8. Дано сторони a , b , c прямокутного паралелепіпеда. Обчислити об'єм паралелепіпеда $V = abc$ та довжину його діагоналі $d = \sqrt{a^2 + b^2 + c^2}$.

9. Дано сторону a тетраедра. Обчислити об'єм тетраедра $V = \frac{a^3 \sqrt{2}}{12}$, площу його поверхні $S = a^2 \cdot \sqrt{3}$ та радіус описаної сфери $R = \frac{a\sqrt{6}}{4}$.

10. Дано радіус основи R та висоту H прямого кругового циліндра. Обчислити об'єм $V = \pi R^2 H$, площу бічної поверхні $S_b = 2\pi R H$ та площу повної поверхні $S_p = S_b + 2\pi R^2$.

11. Дано радіус основи R , висоту H та твірну L прямого кругового конуса. Обчислити об'єм $V = \pi R^2 H / 3$, площу бічної поверхні $S_b = 2\pi R L$ та площу повної поверхні $S_p = S_b + \pi R^2$.

12. Дано радіуси R_1 верхньої та нижньої R_2 основ, висоту H та твірну L зрізаного конуса. Обчислити об'єм $V = \pi(R_1^2 + R_1 R_2 + R_2^2)H / 3$, площу бічної поверхні $S_b = 2\pi(R_2 - R_1)L$ та площу повної поверхні $S_p = S_b + \pi(R_1^2 + R_2^2)$.

13. Дано радіус R кулі. Обчислити об'єм кулі $V = \frac{4\pi R^3}{3}$ та площу сфери $S = 4\pi R^2$.

14. Дано координати двох точок: $M_1(x_1, y_1, z_1)$, $M_2(x_2, y_2, z_2)$, та деякий коефіцієнт k . Обчислити координати точки $M_0(x_0, y_0, z_0)$, яка ділить відрізок M_1M_2 у відношенні k за такими формулами: $x_0 = \frac{x_1 + kx_2}{1+k}$; $y_0 = \frac{y_1 + ky_2}{1+k}$; $z_0 = \frac{z_1 + kz_2}{1+k}$.

15. Дано координати вершин трикутника: $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Обчислити довжини сторін цього трикутника:

$$l_{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}; \quad l_{AC} = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}; \\ l_{BC} = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}.$$

16. Дано ребро a куба. Обчислити об'єм куба $V = a^3$ та площу його бічної поверхні $S = 6a^2$.

17. Дано координати двох точок: $M_1(x_1, y_1)$, $M_2(x_2, y_2)$, та рівняння прямої $Ax + By + C = 0$. Обчислити відстань від заданих точок до заданої прямої за формулою $d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$, де x_0, y_0 - координати точки.

18. Дано координати точки $M(x, y)$ на еліпсі та число c . Обчислити фокальні радіуси точки $F_1M = \sqrt{(x + c)^2 + y^2}$, $F_2M = \sqrt{(x - c)^2 + y^2}$.

19. Дано координати двох векторів $\vec{a}(x_1, y_1)$ та $\vec{b}(x_2, y_2)$. Обчислити скалярний добуток цих векторів $\vec{a}\vec{b} = x_1 * y_1 + x_2 * y_2$ та їх модулі $|\vec{a}| = \sqrt{x_1^2 + y_1^2}$; $|\vec{b}| = \sqrt{x_2^2 + y_2^2}$.

20. Дано сторону a рівностороннього трикутника. Обчислити його площу $S = \frac{a^2 * \sqrt{3}}{4}$, радіус вписаного кола $r = \frac{a\sqrt{3}}{6}$ та радіус описаного кола $R = \frac{a\sqrt{3}}{3}$.

21. Дано радіус R круга. Обчислити площу цього круга $S = \pi R^2$ та довжину кола $l = 2\pi R$, що його обмежує.

22. Дано сторони a, b, c трикутника. Знайти площу цього трикутника за формулою Герона $S = \sqrt{p(p-a)(p-b)(p-c)}$, та радіус вписаного кола $r = \frac{\sqrt{p(p-a)(p-b)(p-c)}}{p}$, де $p = \frac{a+b+c}{2}$ - півпериметр.

23. Дано сторони a, b, c трикутника. Обчислити величину висоти трикутника, опущену на сторону c : $h_c = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{ac}$ та довжину медіани, проведеної до цієї ж сторони $m_c = \frac{\sqrt{2b^2 + 2c^2 - a^2}}{2}$.

24. Дано сторону a октаедра. Обчислити його об'єм $V = \frac{a^3\sqrt{2}}{3}$ та площу поверхні $S = 2a^2\sqrt{3}$.

25. Дано сторону a октаедра. Обчислити радіус описаної сфери $R = \frac{a\sqrt{2}}{2}$ та радіус вписаної сфери $r = \frac{a\sqrt{6}}{26}$.

26. Дано сторони a, b, c трикутника. Обчислити бісектриси кутів, що лежать проти відповідних сторін трикутника за формулами:

$$l_a = \frac{2\sqrt{bcp(p-a)}}{b+c}; \quad l_b = \frac{2\sqrt{acp(p-b)}}{a+c}; \quad l_c = \frac{2\sqrt{abp(p-c)}}{a+b}; \quad \text{де } p = \frac{a+b+c}{2} - \text{півпериметр.}$$

27. Дано сторони a , b , c трикутника. Обчислити бісектриси кутів, що лежать проти відповідних сторін трикутника за формулами:

$$l_a = \frac{\sqrt{bc((b+c)^2 - a^2)}}{b+c}; \quad l_b = \frac{\sqrt{ac((a+c)^2 - b^2)}}{a+c}; \quad l_c = \frac{\sqrt{ab((a+b)^2 - c^2)}}{a+b}$$

28. Дано сторону a рівностороннього трикутника. Обчислити радіуси вписаного $r = \frac{a\sqrt{3}}{6}$ та описаного $R = 2r$ кіл та площу $S = \frac{\sqrt{3}a^2}{4}$ трикутника.

29. Дано основу a та бічну сторону b рівнобедреного трикутника. Обчислити висоту трикутника $h_a = \frac{\sqrt{4b^2 - a^2}}{2}$, опущену на його основу, радіуси вписаного $r = \frac{a\sqrt{4b^2 - a^2}}{2(2b+a)}$ та описаного $R = \frac{b^2}{\sqrt{4b^2 - a^2}}$ кіл та площу $S = \frac{a\sqrt{4b^2 - a^2}}{4}$ трикутника.

30. Дано гіпотенузу c рівнобедреного прямокутного трикутника. Обчислити його катети $a = b = \frac{c\sqrt{2}}{2}$ та площу $S = \frac{c^2}{4}$.

ЛАБОРАТОРНА РОБОТА №2: “ПРОГРАМУВАННЯ ЛІНІЙНИХ ПРОЦЕСІВ, ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ ЗАСОБАМИ МОВИ C++”

МЕТА РОБОТИ

1. Ознайомитися з концепцією потокового введення та виведення даних різних типів засобами мови C++.
2. Ознайомитися з інструментами потокового введення та виведення даних різних типів засобами мови C++.
3. Ознайомитися з маніпуляторами потоків мови C++ і особливостями їх застосування у процесах введення виведення даних різних типів.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_02_01` для розв’язування наступної задачі. Дано: R - радіус основи прямого кругового циліндра; h - висота прямого кругового циліндра. Необхідно обчислити та вивести на друк: $sb=2\pi r h$ - площу бічної поверхні циліндра; $sp=sb+2\pi r^2$ - площу повної поверхні циліндра; $v=\pi r^2 h$ - об'єм циліндра. Результати розрахунків подати у вигляді таблиці. Завдання виконати мовою C++.

```
/* **** */
/* lab_work_02_01.                               */
/* Лінійні обчислювальні процеси засобами мови C++ */
/* Дано: R- радіус основи прямого кругового        */
/* циліндра; h - висота прямого кругового циліндра. */
/* Обчислити та вивести на друк:                    */
/* sb=2*pi*r*h - площу бічної поверхні циліндра;    */
/* sp=sb+2*pi*r*r - площу повної поверхні циліндра; */
/* v=pi*r*r*h - об'єм циліндра.                    */
/* Результати розрахунків подати у вигляді таблиці. */
/* **** */

#include <iostream.h>
#include <conio.h>
#include <math.h>

int main()
{
    // Оголошення констант та змінних:
    clrscr(); // очищення екрану
    const float pi=3.14159; // оголошення константи pi
    float r; // радіус основи прямого кругового циліндра
    float h; // висота прямого кругового циліндра
    float sb; // площа бічної поверхні циліндра
    float sp; // площа повної поверхні циліндра
    float v; // об'єм циліндра
    // Ввід заданих величин:
    cout<<" Введіть радіус основи циліндра: ";
    cin>>r;
    cout<<" Введіть висоту циліндра: ";
    cin>>h;
    // Обчислення параметрів циліндра:
```

```

sb=2*pi*r*h;
sp=sb+2*pi*r*r*h;
v=pi*r*h*h;
// Друк результатів у вигляді таблиці:
cout.precision(3);
cout<<"\n  Вхідні величини та результати обчислень:";
cout<<"\n  I-----I";
cout<<"\n  I          Вхідні величини:          I";
cout<<"\n  I-----I";
cout<<"\n  I  Радіус основи циліндра      I      "<<r<<"      I";
cout<<"\n  I-----I";
cout<<"\n  I  Висота циліндра            I      "<<h<<"      I";
cout<<"\n  I-----I";
cout<<"\n  I          Результати обчислень:          I";
cout<<"\n  I-----I";
cout<<"\n  I  Бічна поверхня циліндра  I      "<<sb<<"      I";
cout<<"\n  I-----I";
cout<<"\n  I  Повна поверхня циліндра  I      "<<sp<<"      I";
cout<<"\n  I-----I";
cout<<"\n  I  Об'єм циліндра            I      "<<v<<"      I";
cout<<"\n  I-----I\n";

getch();
return 0;
}

```

Результат тестування програми:

```

Введіть радіус основи циліндра: 4
Введіть висоту циліндра: 5
Вхідні величини та результати обчислень:
I-----I
I          Вхідні величини:          I
I-----I
I  Радіус основи циліндра      I      4      I
I-----I
I  Висота циліндра            I      5      I
I-----I
I          Результати обчислень:          I
I-----I
I  Бічна поверхня циліндра  I      125.664      I
I-----I
I  Повна поверхня циліндра  I      628.318      I
I-----I
I  Об'єм циліндра            I      314.159      I
I-----I

```

2. Ввести, відлагодити та протестувати програму обчислення виразів $a=f_1(x, y, z)$ та $b=f_2(x, y, z)$ з використанням бібліотеки арифметичних функцій і зберегти її під назвою *lab_work_02_02*.

$$a = \arccos \frac{y}{2 + y^2 + |yz|} (\sqrt{\pi + x^2 + \sin^2 yz} - \lg \frac{2}{x^2 + y^2 + 1});$$

$$b = e^{-3x^2 + \frac{y}{5} - z} + \sqrt[3]{\frac{\operatorname{tg}^2 x}{1 + \sin \frac{\pi y}{2}}} - \left(\frac{x + y + z}{xyz}\right)^3.$$

```

/*****
/* lab_work_02_02.                                     */
/* Бібліотека арифметичних функцій C++.                 */
/* Дано: x,y,z - змінні дійсного типу.                   */
/* Обчислити значення виразів a=f1(x,y,z) та            */
/* b=f2(x,y,z). Вивести на друк вхідні величини і      */
/* результати обчислень, у тому числі значення        */
/* проміжних величин, у вигляді таблиці.               */
/*                                                     */
/*****
#include <iostream.h>
#include <iomanip.h>
#include <windows.h>
#include <conio.h>
#include <math.h>
int main()
{clrscr();
  SetConsoleCP(1251);
  SetConsoleOutputCP(1251);
  const float pi=3.14159;
  float x,y,z;
  float a1,a2,a3,a4,a5,a;
  float b1,b2,b3,b4,b5,b;
  // Ввід заданих величин:
  cout<<" Введіть значення трьох дійсних змінних:\n ";
  cin>>x>>y>>z;

  // Обчислення виразу a:
  a1=y/(2+y*y+fabs(y*z));
  a2=acos(a1);
  a3=pi+x*x+pow(sin(y*z),2);
  a4=pow(a3,1/2);
  a5=log10(2/(x*x+y*y+1));
  a=a2*(a4-a5);

  // Обчислення виразу b:
  b1=exp(-3*x*x+y/3-z);
  b2=pow(tan(x),2);
  b3=b2/(1+sin(pi*y/2));
  b4=pow(b3,1/3);
  b5=(x+y+z)/(x*y*z);
  b=b1+b4-pow(b5,2);

  // Друк результатів у вигляді таблиці:
  cout<<fixed<<setprecision(3)<<"\n Вхідні величини та результати
обчислень:";

```

```

cout<<"\n I-----I";
cout<<"\n I          Вхідні величини:          I";
cout<<"\n I-----I";
cout<<"\n I          x=          I"<<"          "<<x<<"          I";
cout<<"\n I-----I";
cout<<"\n I          y=          I"<<"          "<<y<<"          I";
cout<<"\n I-----I";
cout<<"\n I          z=          I"<<"          "<<z<<"          I";
cout<<"\n I-----I";
cout<<"\n I          Результати обчислень:          I";
cout<<"\n I-----I";
cout<<"\n I a1= I          "<<a1<<" I          b1= "<<b1<<"          I";
cout<<"\n I-----I";
cout<<"\n I a2= I          "<<a2<<" I          b2= "<<b2<<"          I";
cout<<"\n I-----I";
cout<<"\n I a3= I          "<<a3<<" I          b3= "<<b3<<"          I";
cout<<"\n I-----I";
cout<<"\n I a4= I          "<<a4<<" I          b4= "<<b4<<"          I";
cout<<"\n I-----I";
cout<<"\n I a5= I          "<<a5<<" I          b5= "<<b5<<"          I";
cout<<"\n I-----I";
cout<<"\n I a = I          "<<a<<" I          b = "<<b<<"          I";
cout<<"\n I-----I\n";
getch();
return 0; }

```

Результати тестування програми:

Введіть значення трьох дійсних змінних:

1 2 3

Вхідні величини та результати обчислень:

```

I-----I
I          Вхідні величини:          I
I-----I
I          x=          I          1.00          I
I-----I
I          y=          I          2.00          I
I-----I
I          z=          I          3.00          I
I-----I
I          Результати обчислень:          I
I-----I
I a1= I          0.17 I          b1= I          0.00 I
I-----I
I a2= I          1.40 I          b2= I          2.43 I
I-----I
I a3= I          4.22 I          b3= I          2.43 I
I-----I
I a4= I          1.00 I          b4= I          1.00 I
I-----I
I a5= I          -0.48 I          b5= I          1.00 I
I-----I
I a = I          2.07 I          b = I          0.00 I
I-----I

```

3. Ввести, відлагодити та протестувати програму потокового виведення даних базових типів мови C++ з використанням відповідних маніпуляторів і зберегти її під назвою *lab_work_02_03*.

```
/* **** */
/* lab_work_02_03. */
/* Введення/виведення даних різних типів засобами */
/* мови C++. */
/* Дано: змінні різних типів: */
/* char ac='A', bc='B', cc, dc; */
/* int ai=18, bi=-18, ci, di; */
/* unsigned int au=18, bu=24, cu, du; */
/* float af=12.34567,bf=12.234567,cf,df; */
/* Необхідно: */
/* Створити програму мовою C++, у якій передбачити */
/* введення значень змінних, які не ініціалізовані, */
/* та виведення значень всіх змінних з використанням */
/* маніпуляторів для форматування даних */
/* **** */
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>

char ac='A', bc='B', cc, dc;
int ai=18, bi=-18, ci, di;
unsigned int au=18, bu=24, cu, du;

float af=12.34567,bf=12.234567,cf,df;

int main () {
    clrscr();

    //Введення/виведення даних символьного типу
    cout<<"\n Введіть значення змінних cc та dc символьного типу:"
<<endl;
    cin>>cc>>dc;
    cout<<"\n Значення змінної ac="<<setw(10)<<right<<ac<<endl;
    cout<<"\n Значення змінної bc="<<setw(10)<<left<<bc<<endl;
    cout<<"\n Значення змінної cc="<<setw(10)<<right<<cc<<endl;
    cout<<"\n Значення змінної dc=";
    cout.width(10);
    cout<<left<<dc<<endl<<endl;

    // Введення/виведення даних цілого типу зі знаком
    cout<<"\n Введіть значення змінних ci та di цілого типу зі знаком:"
<<endl;
    cin>>ci>>di;
    cout<<"\n Значення змінної ai="<<setw(10)<<right<<ai<<endl;
    cout<<"\n Значення змінної bi="<<setw(10)<<right<<bi<<endl;
    cout<<"\n Значення змінної ci="<<setw(10)<<left<<ci<<endl;
    cout<<"\n Значення змінної di="<<setw(10)<<left<<di<<endl<<endl;
```

```

// Введення/виведення даних цілого типу без знаку
cout<<"\n Введіть значення змінних cu та du цілого типу без знаку:
"<<endl;
cin>>cu>>du;
cout<<"\n Значення змінної au=\n"
<<setw(5)<<dec<<right<<au<<" - в 10-й нотації \n"
<<setw(5)<<oct<<right<<au<<" - у 8-й нотації \n"
<<setw(5)<<hex<<right<<au<<" - в 16-й нотації "<<endl<<endl;
cout<<"\n Значення змінної bu=\n"
<<setw(5)<<dec<<left<<bu<<" - в 10-й нотації \n"
<<setw(5)<<oct<<left<<bu<<" - у 8-й нотації \n"
<<setw(5)<<hex<<left<<bu<<" - в 16-й нотації n"<<endl<<endl;
cout<<"\n Значення змінної cu=\n"
<<setw(5)<<dec<<right<<cu<<" - в 10-й нотації \n"
<<setw(5)<<oct<<right<<cu<<" - у 8-й нотації \n"
<<setw(5)<<hex<<right<<cu<<" - в 16-й нотації "<<endl<<endl;
cout<<"\n Значення змінної du=\n"
<<setw(5)<<dec<<left<<du<<" - в 10-й нотації \n"
<<setw(5)<<oct<<left<<du<<" - у 8-й нотації \n"
<<setw(5)<<hex<<left<<du<<" - в 16-й нотації "<<endl<<endl;

// Введення/виведення даних дійсного типу
cout<<"\n Введіть значення змінних cf та df дійсного типу:
"<<endl;
cin>>cf>>df;
cout<<setprecision(4);
cout<<"\n Значення змінної af="<<right<<fixed<<af<<endl;
cout<<"\n Значення змінної bf="<<right<<fixed<<bf<<endl;
cout<<"\n Значення змінної cf="<<left<<scientific<<cf<<endl;
cout<<"\n Значення змінної df="<<left<<scientific<<df<<endl;

cout<<"\n Значення змінної af="<<setw(12)<<showpoint<<
setfill('X')<<right<<fixed<<af<<endl;
cout<<"\n Значення змінної af="<<setw(12)<<showpoint<<
setfill('X')<<left<<fixed<<af<<endl;
cout<<"\n Значення змінної af="<<setw(12)<<showpoint
<<setfill('X')<<right<<scientific <<af<<endl;
cout<<"\n Значення змінної af="<<setw(12)<<showpoint
<<setfill('X')<<left<<scientific <<af<<endl;

getch();
return 0;
}

```

Результати тестування програми:

Введіть значення змінних ss та ds символного типу:

C D

Значення змінної ac=	A
Значення змінної bc=	B
Значення змінної ss=C	
Значення змінної ds=D	

Введіть значення змінних ci та di цілого типу зі знаком:
123 -123

Значення змінної ai= 18
Значення змінної bi= -18
Значення змінної ci=123
Значення змінної di=-123

Введіть значення змінних ci та di цілого типу без знаку:
124 421

Значення змінної au=
18 - in 10-й нотації
22 - in 8- й нотації
12 - in 16- й нотації

Значення змінної bu=
24 - in 10- й нотації
30 - in 8- й нотації
18 - in 16- й нотації

Значення змінної cu=
124 - in 10- й нотації
174 - in 8- й нотації
7c - in 16- й нотації

Значення змінної du=
421 - in 10- й нотації
645 - in 8- й нотації
1a5 - in 16- й нотації

Введіть значення змінних cf та df дійсного типу:
35.3456789 -35.3456789

Значення змінної af=12.3457
Значення змінної bf=12.2346
Значення змінної cf=3.5346e+01
Значення змінної df=-3.5346e+01

Значення змінної af=XXXXX12.3457
Значення змінної af=12.3457XXXXX
Значення змінної af=XX1.2346e+01
Значення змінної af=1.2346e+01XX

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати лінійну програму обчислення значень змінних x та y при заданих значеннях аргументів x, y та z, вибраної згідно з варіантом.

1.
$$a = \frac{\sqrt{|x-1|} + \sqrt[3]{y^2+1}}{1 + \frac{x^2}{2} + \frac{y^2}{4}} \sin x + \ln(1+x^4) - e^{2x-y};$$

$$b = \frac{x(\operatorname{arctg} z - \lg(\frac{4}{1+x^2}))}{\pi + \cos^2(y+z^2)}.$$

$$2. \quad a = \left(x \operatorname{ctg} \frac{\pi}{5} + \sqrt{x^2 + y^2 + 4} \right) \sin^2 x + \ln \frac{1+z^2}{3.5+x^2+y^2} + e^{-|x|} \operatorname{arctg} \frac{z}{2};$$

$$b = 78.06x^2 + \frac{y}{1.2x^2 + \frac{2 + \sin x}{\sqrt[3]{5.1 - |x^2 - 1|}}}.$$

$$3. \quad a = \left(\frac{x+1}{x^2+1} + 4.1 \left(\frac{x^2+3x-1}{x^2+1} \left(\sin^2 x + \frac{1}{2} \right) \right)^2 + e^{y-x} \lg(y^2 + 1) \right)^3;$$

$$b = \sqrt{\left| \frac{1.2z - 0.5y^2}{1 + tg^2 \frac{\pi}{8}} \right|} + 2 + \frac{\operatorname{arctg} z}{2.5 + \sin^2(xyz)}.$$

$$4. \quad a = e^{-xyz} \left(\lg \left(x^2 + \cos y + \operatorname{arctg} \frac{z^2}{4} + 3 \right) + \sqrt[3]{(y^2 + z^2 - 3xy + 1.1)^2 + 0.8} \right);$$

$$b = \sin z + \sqrt{1 + 3.1|x^2 - y^2|} \frac{1 + \sqrt{|x| + 2} + (y + z)^2}{(2.5 + x^3)(1 + \cos^2 z)}.$$

$$5. \quad a = \frac{1.7 + \ln(4 + x^2) + \operatorname{arctg} z - 2.1xy}{2 + \left| \frac{\sqrt{|x-y|+1} + 3xyz}{\pi + \sin(x+z) + \cos^2 \frac{y}{x^2+y^2+1}} \right|};$$

$$b = \sqrt[3]{\left| \frac{\lg(1 + x^2 + y^2 + z^2) + e^{-2.1+x+y}}{\frac{3}{4} \left(1 + \frac{x^2}{2} + \frac{y^2}{4} + \frac{z^2}{8} \right) (0.1 + \cos^2(x-y))} \right|}.$$

$$6. \quad a = 3x^2y^3z + \ln(4 + (x-y)^2) - \operatorname{arctg}(3\pi + 1.2xy - \frac{z}{4});$$

$$b = e^{-1+x} (2x^2 + 0.4y + \sqrt[3]{z^2 + 1}) (\cos^2 \left(\frac{\pi}{8} + x \right) + \sin^2 \left(\frac{y}{3.2} + z \right) + 3)^2.$$

$$7. \quad a = \frac{x + e^{-x+1.3} \ln(1+x^2+y^2) + \frac{3}{4}y + \pi z}{(8 + \frac{x^2}{2} + \frac{y^2}{3} + z)(\cos^2 x + \sin^2(y+z) + 1.001)};$$

$$b = (1.8 \frac{\sqrt{|x-y+z|+0.7}}{y^2+9} + \log \operatorname{tg} \frac{\pi}{5})^2 + \sin \frac{y-z}{1+x^2} \times \frac{xyz+1.2}{e^{x+2y}}.$$

$$8. \quad a = (y + 1) \frac{e^{-\frac{1}{2}|x^2-z|+\ln(3+x^2)+|x-y+2z|}}{4+\sin^2\left(\pi-\frac{x}{2}\right)+\cos^2\left(x+\frac{y}{2}-\frac{z}{3}\right)};$$

$$b = \sqrt{\lg\left(10 + \frac{1+\arctg(x^2-y^2)+1.8\pi yz}{(5.41+\frac{1}{3}x^2+y^2)(1+|x+y+z|)}\right)}.$$

$$9. a = \left(\cos^2(\arctg z) + 4.2e^{-x^2+1.3} - \sqrt{2x^2 + y^2 + 1} + \ln\left(\frac{2+3.3x^2+y^2}{4x^2+5.5y^2+1}\right)\right);$$

$$b = \sin\left(x + \frac{\pi}{4}\right) \lg\left(1 + \sqrt{4 + x^2 + \frac{y^2}{2}}\right) + \operatorname{tg}(1 + x^2);$$

$$10. b = \operatorname{tg}^2 \frac{\pi+2x-\frac{yz}{3}}{1+|y^2-z^3|} e^{\frac{3.2x+y}{4}} + \frac{xy+xz+yz}{3} - \sin^2\left(\pi x + \frac{y}{3} + \frac{z}{4}\right);$$

$$b = \frac{(2.4\arctg(x+y)+(x-y)^2+1.3\cos^2+\pi y)^2}{\left(1.5+\ln\frac{1+|x|}{2.1+\cos z}\right)(4.6+y^2+z^2+\sqrt{|xyz|})}.$$

$$11. \quad a = \frac{\sqrt{|x=1|+8}+4.1\arctgz+\sin^2\pi(x-y)-2y+z^3}{\left(1+\left|y-\operatorname{tg}z\right|(2+x^2+\frac{y^2}{3.14+z^2}+\cos^3y)\right)};$$

$$b = \frac{\ln(5+|x^2-3|+\sqrt[3]{\frac{xy}{z}})}{2+\frac{\lg|2+x|+e^{-y^2}+z}{3+\frac{x^2}{2}+\frac{y^4}{4}+\frac{z^6}{6}}}.$$

$$12. \quad a = \lg\sqrt[1]{\frac{x^2+\pi|x-y|}{e^{y+z+0.2x^2+3}}} + \sin^2\left(\frac{x}{yz}\right);$$

$$b = \left(\cos\frac{x}{1+x^2+y^2} - \arctg\frac{y^2+z^2}{1+|xyz|}\right)^3 + 1n(4 + \sin^2 y).$$

$$13. \quad a = \left(\frac{x+2y+3z}{1+|x-y|}\right)^2 (\ln(2 + \cos^2 \pi x) e^{-(x^2+y^2)})^3;$$

$$b = \operatorname{tg} \frac{\pi}{8} + \frac{\sin^3(2x - \frac{3}{2y + \pi z})}{y^2 + \frac{x^2 - y^2}{1 + 2.2z^2}}.$$

$$14. \quad a = \frac{x+1.3y+4.8z^2-\frac{1}{5}}{x^2+2y^2+\pi+5z^2} + \frac{e^{-(x-y)} \ln(4.2+\operatorname{arctg} z)}{\cos^2 x + \sin^2 y + 0.614};$$

$$b = \sqrt{\ln \left(x + \frac{\sin x + 0.2z^2 + e^{-\frac{x^2}{2}}}{2+|3x-4y+5z|} \right)}.$$

$$15. \quad a = \sin^2 \frac{\ln(1+|x-y-z|) - \sqrt{2y^2 + \pi z^2}}{0.5+2x^2+3y^2+z^2};$$

$$b = (e^{-\frac{1}{3x}+2y^2} \operatorname{arctg} \frac{4z}{1+x^2+y^2} + \cos^2 \frac{\pi x}{3+\frac{z^2}{2}})^{\frac{1}{3}}.$$

$$16. \quad a = \frac{3.41-x^2-1.2xy+5\sin^2 \pi z + \sqrt{|x^2-xy|}}{e^{-(x^2+y^2+\frac{z}{2})} + \operatorname{arctg}(3.3x+y-9.1z)};$$

$$b = \sqrt{9 + \left| \frac{x^2 + \operatorname{arctg} z + y(4-x^2)^3}{\pi + \sin^2 x + \cos^2 y + \sin z} \right|}$$

$$17. \quad a = \sin^2 \frac{\pi + \sin^2 z - 4.01y + \ln(1+|xyz|)}{\operatorname{arctg} \frac{z}{4} + \sqrt{25.05 + e^{-\frac{x}{2}} + 0.1y^2 + z^2}};$$

$$b = \frac{2x-3.4}{y^2+1} + 4.9 \left(\left(\frac{1}{3} + \cos^2(x+y) \right) \frac{2z-x^2}{1+|x-y|} \right)^3 + e^{-x} \lg(2+|x|).$$

$$18. \quad a = \left(\frac{\sin^2(x+y) - 0.17z}{1+\ln(2+\cos^3 x)} + e^{-\pi(\frac{x}{2}+y^2)} + \sqrt[3]{(4x+1.3y-z)^2} \right)^3;$$

$$b = \operatorname{arctg} \frac{z}{1+x^2} + \frac{7x - \frac{4}{3} + \operatorname{tg} \frac{\pi z}{5} + x^4}{2 + \frac{x^2 - y^2 + x \sin y}{4 + \sqrt{2+x^2+y^2}}}.$$

$$19. \quad a = \cos^2 \frac{\pi - \ln \left(1 + \frac{3}{x^2+2} \right) + \left| x + \frac{y}{2} \right| - (x-4.2)(y+2.8)}{(2x^2+3y^2+\ln(1+|z|)) \sin((x+y)^2+1)};$$

$$b = \sqrt{\lg \left(25 + \frac{3y^2+x^2+1.43}{4+\sin \frac{x}{2}} \right) \left(4 + \operatorname{arctg} \frac{y-0.1x}{z^2+8} \right)};$$

$$20. \quad a = \left(\frac{x^2}{2} + \frac{y^2}{4} + 3.3xy\right)^3 \ln \sqrt{e^{-1.2x} + \cos^2 \frac{x-y}{x^2+2}} + \pi;$$

$$b = tg \frac{7x - |y - z| + 3.2yz}{x^2 + 2y^2 + 8.41z^2} \left(\frac{x^2}{2} + \frac{y^3}{3} + lg(3 + |yz|)\right)^{\frac{3}{2}}$$

$$21. \quad a = yctg \frac{\pi}{7} + \sqrt{(2.1x^2 + 0.5y^2 + 2)\cos^4 x + e^{-|x+1|}arctgz};$$

$$b = \sin \frac{x+y+z}{4+z} + 3.4\left(\left(\frac{1}{2} + \cos^2 \frac{y}{3}\right) \frac{tg \frac{\pi}{3} + 1 + \sqrt{|x|}}{1 + lg(4 + |xyz|)}\right)^3.$$

$$22. \quad a = (3.2x + lg(1 + |y|))^3 lg \left(\frac{8 + arctg(y+z)}{1+x^2+y^4+\sin^2 x}\right);$$

$$b = e^{-\left|\frac{x-y-0.5z}{ctg \frac{\pi}{7}}\right|} \ln^3 \left(2 + \frac{x^2}{4} + \frac{y^2}{6} + tg^2 \frac{x}{yz}\right).$$

$$23. \quad a = \sqrt[3]{arctg \left(\frac{1.4y-0.5z^2}{x-2y+3z}\right)} + \ln \frac{x^2}{1.3+y^2} + tg \frac{1}{xyz};$$

$$b = e^{-\frac{lg(1+x^2)(\sin^2 x + \cos \frac{y}{2})}{4+x^2+y^2+z^2(1-|xy|)}}.$$

$$24. \quad a = \cos \frac{(x+y+z)^2 + \ln(2+|xy|+z^2) + arctg z}{\frac{x^2}{4} + \frac{y^2}{8} + z^2 + 1.5};$$

$$b = e^{3.2 + \sin x} \left(tg \frac{\pi}{8} + \sqrt[3]{1 + \sin^2 z + \cos^2 y} \frac{3xyz}{lg \left(1 + \frac{y^2}{2}\right)} \right).$$

$$25. \quad a = \left(\frac{2x}{1+x^2+y^2} + \cos xy + \ln \frac{1+|x|\sin^2 x}{3+x^2+y^2}\right)^3 tg \frac{\pi(y+\frac{1}{2})}{25};$$

$$b = e^{\frac{2.5x^2+y}{1+x^2+y^2}} lg \sqrt{1 + \sin^2 x + \cos^2(y-z)} arctg \frac{x+y+z}{3}$$

$$26. a = \left(\left(\frac{x+y+z}{xyz} \right) + \sqrt[3]{|x| + \cos^2 y + \operatorname{tg} z} \right) e^{-\sin^2 \pi x} + \ln(1.1 + \cos \frac{z}{4});$$

$$b = \arcsin(z+x) e^{-x^3+y^3} \sqrt{2 + \lg \cos(2+xy)} + 1.154.$$

$$27. a = \left(\lg \frac{x^2}{1+3.5z^4} - \sqrt[4]{2 - \sin x - \operatorname{tg}(y+z)} \right) \left(1 + \sqrt{|x|} \right);$$

$$\left(b = \arccos \frac{z}{x^2+z^2} + e^{-3.4+|xy|} \right) \left(\sin^2 x + \cos^3 yz + \frac{1}{4} \right).$$

$$28. a = \left(\operatorname{tg} \frac{x+y+z}{3+x^2} - \sqrt[3]{3x - 4y^2 + 5z^3} \right) |4 + \sqrt{\lg |x| + 2}|;$$

$$b = \left(\arcsin \frac{x-y}{x^2+y^2} + \sqrt{1 + |x| + z^2} \right) e^{-\frac{\sin \pi x}{3+z^3}};$$

$$29. a = e^{-\frac{\operatorname{tg} x}{\left(\frac{x^2+y^2}{xz} \right)}} \left(\left| \frac{\sqrt[3]{3+xyz} - (x+y)^4}{2 + \sin 2x + \lg(2+|yz|)} \right| - \frac{xyz}{3+x^2+y^4z^2} \right);$$

$$b = \sqrt[3]{\frac{1+x \cos 2|xyz|}{\ln^2(3+z^2)}} + \arcsin \frac{z}{1+z^2+|xyz|} - 2.364.$$

$$30. a = \arccos \frac{y}{2+y^2+|yz|} \left(\sqrt{3.14 + x^2 + \sin^3 yz} - \lg \frac{2}{x^2+y^2+1} \right);$$

$$b = e^{-3x^2+\frac{y}{5}-z} + \sqrt[3]{\frac{\operatorname{tg}^2 x}{1 + \sin \frac{\pi y}{2}}} - \left(\frac{x+y+z}{xyz} \right)^3.$$

ЛАБОРАТОРНА РОБОТА №3: “ФУНКЦІЇ КОРИСТУВАЧА З ПАРАМЕТРАМИ-ЗНАЧЕННЯМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Ознайомитися з технологією створення функцій користувача з параметрами-значеннями;

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_03_01` для розв’язування наступної задачі. Дано: R - радіус основи прямого кругового циліндра; h - висота прямого кругового циліндра. Необхідно обчислити та вивести на друк: $sb=2\pi r h$ - площу бічної поверхні циліндра; $sp=sb+2\pi r^2$ - площу повної поверхні циліндра; $v=\pi r^2 h$ - об’єм циліндра. Завдання виконати мовою C з використанням функцій користувача з параметрами-значеннями.

```
/* **** */
/* lab_work_03_01. */
/* Підпрограми-функції з параметрами-значеннями */
/* засобами мови C. */
/* Дано: R- радіус основи прямого кругового */
/* циліндра; h - висота прямого кругового циліндра. */
/* Обчислити та вивести на друк: */
/* sb=2*pi*r*h - площу бічної поверхні циліндра; */
/* sp=sb+2*pi*r*r - площу повної поверхні циліндра; */
/* v=pi*r*r*h - об'єм циліндра. */
/* Для обчислення величин sb, sp та v скласти */
/* підпрограми-функції з параметрами-значеннями. */
/* **** */
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define pi 3.14159

// Прототипи функцій:
float BitshPoverchnja(float rc, float hc);
float PovPoverchnja(float rc, float hc);
float Objem(float rc, float hc);

int main()
{clrscr();

// Оголошення констант та змінних:
float r; // радіус основи прямого кругового циліндра
float h; // висота прямого кругового циліндра
float sb; // площа бічної поверхні циліндра
float sp; // площа повної поверхні циліндра
float v; // об'єм циліндра
```

```

// Ввід заданих величин радіуса та висоти:
printf ("\n Введіть радіус основи циліндра: ");
scanf("%f",&r);
printf ("\n Введіть висоту циліндра: ");
scanf("%f",&h);

// Обчислення параметрів циліндра:
printf ("\n Площа бічної поверхні циліндра sb=%f\n",
        BitshPoverchnja(r,h));
printf (" Площа повної поверхні циліндра sp=%f\n",
        PovPoverchnja(r,h));
printf (" Об'єм циліндра                                v=%f\n",
        Objem(r,h));
getch();
return 0;
}

//-----
// Підпрограма-функція обчислення площі бічної поверхні
float BitshPoverchnja(float rc, float hc) {
    return 2*pi*rc*hc;
}

//-----
// Підпрограма-функція обчислення площі повної поверхні
float PovPoverchnja(float rc, float hc) {
    return BitshPoverchnja(rc,hc)+2*pi*rc*rc;
}

//-----
// Підпрограма-функція обчислення об'єму циліндра
float Objem(float rc, float hc) {
    return pi*rc*rc*hc;
}

```

Результати тестування програми:

Введіть радіус основи циліндра: 4
Введіть висоту циліндра: 9

Площа бічної поверхні циліндра sb=226.194480
Площа повної поверхні циліндра sp=326.725360
Об'єм циліндра v=452.388960

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати лінійну програму обчислення значень змінних a та b при заданих значеннях аргументів x , y та z відповідно до завдання лабораторної роботи №2, склавши для обчислення допоміжних величин a_1 , a_2 , і т.д. та b_1 , b_2 , і т.д. функції з параметрами-значеннями.

ЛАБОРАТОРНА РОБОТА №4: “ФУНКЦІЇ КОРИСТУВАЧА З ПАРАМЕТРАМИ-ВКАЗІВНИКАМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Ознайомитися з технологією створення функцій користувача з параметрами-вказівниками.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_04_01` для розв’язування наступної задачі. Дано: R - радіус основи прямого кругового циліндра; h - висота прямого кругового циліндра. Необхідно обчислити та вивести на друк: $sb=2\pi r h$ - площу бічної поверхні циліндра; $sp=sb+2\pi r^2$ - площу повної поверхні циліндра; $v=\pi r^2 h$ - об’єм циліндра. Завдання виконати мовою C з використанням функцій користувача з параметрами-вказівниками.

```
/* **** */
/* lab_work_04_01. */
/* Функції користувача з параметрами-вказівниками */
/* засобами мови C. */
/* Дано: R- радіус основи прямого кругового */
/* циліндра; h - висота прямого кругового циліндра. */
/* Обчислити та вивести на друк: */
/* sb=2*pi*r*h - площу бічної поверхні циліндра; */
/* sp=sb+2*pi*r*r - площу повної поверхні циліндра; */
/* v=pi*r*r*h - об'єм циліндра. */
/* Для обчислення величин sb, sp та v скласти */
/* функцію користувача з параметрами-вказівниками. */
/* **** */
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define pi 3.14159

// Прототип підпрограми-функції:
int cylindr(float rc, float hc, float *Psb, float *Psp, float *Pv);

int main()
{clrscr();

// Оголошення констант та змінних:
float r; // радіус основи прямого кругового циліндра
float h; // висота прямого кругового циліндра
float sb; // площа бічної поверхні циліндра
float sp; // площа повної поверхні циліндра
float v; // об'єм циліндра

// Ввід заданих величин радіуса та висоти:
printf ("\n Введіть радіус основи циліндра: ");
scanf ("%f",&r);
```

```

printf ("\n Введіть висоту циліндра: ");
scanf ("%f", &h);

// Обчислення параметрів циліндра:
cylindr(r,h, &sb, &sp, &v);

printf ("\n Площа бічної поверхні циліндра sb=%f\n", sb);
printf (" Площа повної поверхні циліндра sp=%f\n", sp);
printf (" Об'єм циліндра v=%f\n", v);

getch();
return 0;
}

//-----
// Підпрограма-функція з параметрами-вказівниками
// для обчислення параметрів циліндра

int cylindr(float rc, float hc, float *Psb, float *Psp, float *Pv) {
    *Psb=2*pi*rc*hc;          // Площа бічної поверхні
    *Psp=*Psb+2*pi*rc*rc;     // Площа повної поверхні
    *Pv=pi*rc*rc*hc;          // Об'єм циліндра
    return 1;
}

```

Результати тестування програми:

```

Введіть радіус основи циліндра: 4
Введіть висоту циліндра: 9
Площа бічної поверхні циліндра sb=226.194473
Площа повної поверхні циліндра sp=326.725342
Об'єм циліндра v=452.388947

```

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати лінійну програму розв'язування задачі, заданої у лабораторній роботі №1, склавши для обчислення вихідних величин функцію з параметрами-вказівниками.

ЛАБОРАТОРНА РОБОТА №5: “ПРОГРАМУВАННЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ ОПЕРАТОРА IF НА ПРИКЛАДІ ОБЧИСЛЕННЯ ЗНАЧЕННЯ СКЛАДНОЇ ФУНКЦІЇ”

МЕТА РОБОТИ

1. Вивчити оператор if мови C/C++.
2. Вивчити команду ? мови C/C++.
3. Ознайомитися з технологією створення розгалужених програм засобами мови C/C++.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою *lab_work_05_01* для розв’язування наступної задачі. Обчислити значення складної функції

$$y = \begin{cases} x^2 - 3x + 1, & x < -2; \\ \sin^2(3x) \cdot \log_3|2x - 1|, & -2 \leq x \leq 0; \\ \sqrt{x^2 + 5x + 1}, & 0 < x < 2; \\ 2^{-x}, & x \geq 2, \end{cases}$$

якщо аргумент x - довільне дійсне число.

```
/* **** */
/* lab_work_06_01. */
/* Обчислення значення складної функції засобами мови C. */
/* Дано: x - Значення аргумента складної функції - довільне */
/* дійсне число; */
/* Обчислити: */
/* Значення складної функції, заданої різними виразами на */
/* різних інтервалах; */
/* y= x*x-3*x+1, якщо x<-2; */
/* y=sin^2(3*x)*log3|2x-1|, якщо -1<=x<=0; */
/* y= sqrt(x*x+5*x+1), якщо 0<x<2; */
/* y=2^-x, якщо x>=2. */
/* функція не існує, в інших випадках. */
/* Створити функцію float f(float x) з параметром-значенням */
/* для обчислення значення заданої функції. */
/* Реалізувати програму мовою C. */
/* **** */
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define pi 3.14159
//Оголошення змінних:
float x; // значення аргумента функції
float y; // значення функції (його потрібно обчислити)

// Прототип підпрограми-функції:
float f(float x); // параметр значення,

int main() {
    clrscr();
```

```

// Введення значення аргумента (довільне дійсне число):
printf ("\n Введіть значення аргумента функції: ");
scanf ("%f",&x);

// Обчислення значення функції через звертання до
//підпрограми-функції:
y=f(x);
printf ("\n Значення аргумента x= %f. Значення функції y=%f\n",
x,y);

getch();
return 0;
}
//-----
// Підпрограма-функція, яка обчислює значення функції уу
float f(float x) {
    float уу; // Значення, яке обчислюється підпрограмою-функцією
    if (x<-2)
        уу= x*x-3*x+1;
    else if ((x>=-2) && (x<=0))
        уу=sin(3*x)*sin(3*x)*log(fabs(2*x-1))/log(3);
    else if ((x>0) && (x<2))
        уу= x*x+5*x+1;
    else
        уу=pow(2, (-x));
    return уу;
}

```

Результати виконання програми:

```

Введіть значення аргумента: -4
x=-4.00      y=25.00000

```

```

Введіть значення аргумента: 0
x=1.00      y=0.000000

```

```

Введіть значення аргумента: 1
x=1.00      y=2.645771

```

```

Введіть значення аргумента: 2
x=2.00      y=0.250000

```

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати розгалужену програму для обчислення значення складної функції $y=f(x, a, b)$ за допомогою умовного оператора `if` при заданих значеннях змінної x та параметрів a і b .

№ варіанту	Функція	Змінні	Межі
1	$y = \begin{cases} \frac{e^{x+4}}{\cos(6x)}, & x \leq 0 \\ \log \sqrt{x}, & 0 < x < 3 \\ x^3, & x \geq 3 \end{cases}$	e – стала (2.718); x, y - змінні	$x \leq 0$ $0 < x < 3$ $x \geq 3$
2	$y = \begin{cases} \sqrt[3]{x^2 + a }, & x \leq 1 \\ \sin^2(ax) * \cos^2(bx), & 1 < x \leq 3 \\ a + x + b , & x > 3 \end{cases}$	a, b - параметри; x, y - змінні	$x \leq 1$ $1 < x \leq 3$ $x > 3$
3	$y = \begin{cases} e^{-x+6}, & x \leq 0 \\ \sqrt[5]{a^2 + 1} + x, & 0 < x < 1 \\ b + \sqrt{x}, & x \geq 1 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x \leq 0$ $0 < x < 1$ $x \geq 1$
4	$y = \begin{cases} \sqrt[4]{ a + bx }, & x < 1 \\ \ln^2(b^2 + x), & 1 \leq x < 2 \\ e^{b+3}, & x \geq 2 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x < 1$ $1 \leq x < 2$ $x \geq 2$
5	$y = \begin{cases} \sin x + \sqrt[4]{ a }, & x < 1 \\ \log b^2 + x , & 2 \leq x < 4 \\ \cos^3(ax), & x \geq 4 \end{cases}$	a, b - параметри; x, y - змінні	$x < 1$ $2 \leq x < 4$ $x \geq 4$
6	$y = \begin{cases} \sqrt[3]{a^2 + 3x^2}, & x < b \\ \ln^2 x - b , & b \leq x \leq b + 1 \\ e^{ax} - x^4, & x > b + 1 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x < b$ $b \leq x \leq b + 1$ $x > b + 1$
7	$y = \begin{cases} \arctg(ax) + \sqrt[3]{ x + 2}, & ax < 1 \\ \sin(bx) + 1.4, & 1 \leq ax \leq 2 \\ \ln^2(x^2 + ab), & ax > 2 \end{cases}$	a, b - параметри; x, y - змінні	$ax < 1$ $1 \leq ax \leq 2$ $ax > 2$
8	$y = \begin{cases} \ln^2 \frac{x^2 + a }{1 + x^4}, & x \leq 3 \\ b^3 + \cos^2(x), & 3 < x \leq 5 \\ \sqrt[3]{ ab x}, & x > 5 \end{cases}$	a, b - параметри; x, y - змінні	$x \leq 3$ $3 < x \leq 5$ $x > 5$
9	$y = \begin{cases} 2\sqrt[3]{\sin(ax + 10)}, & x \leq 4 \\ \ln^2(a + bx + 1), & 4 < x < 5 \\ \frac{x^3}{a^2 + b^2 + 2}, & x \geq 5 \end{cases}$	a, b - параметри; x, y - змінні	$x \leq 4$ $4 < x < 5$ $x \geq 5$
10	$y = \begin{cases} \ln^3(ax + b + 1), & ax < 1 \\ e^{-\sqrt{x^2 + a^2}} * \sin(bx), & 1 \leq ax < 2 \\ \arctg(ax - b), & ax \geq 2 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$ax < 1$ $1 \leq ax < 2$ $ax \geq 2$
11	$y = \begin{cases} \frac{x^2 + a^2}{b^2 + 1}, & x \leq 0 \\ \tan^3(ax), & 0 < x \leq 3 \\ \ln \sqrt[3]{x^2 + 1}, & x > 3 \end{cases}$	a, b - параметри; x, y - змінні	$x \leq 0$ $0 < x \leq 3$ $x > 3$
12	$y = \begin{cases} \cos^3(ax), & bx \leq 1 \\ 2.5 + \frac{e^{-x^2}}{\ln 2 + bx }, & 1 < bx \leq 4 \\ \arctg \sqrt[4]{bx}, & bx > 4 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$bx \leq 1$ $1 < bx \leq 4$ $bx > 4$

13	$y = \begin{cases} \sqrt[3]{ b + \cos^2(ax) }, & ax < 2 \\ \ln^4(ax + b), & 2 \leq ax \leq 5 \\ \ln(x^2 + a^2), & ax > 5 \end{cases}$	a, b - параметри; x, y - змінні	$\begin{aligned} ax < 2 \\ 2 \leq ax \leq 5 \\ ax > 5 \end{aligned}$
14	$y = \begin{cases} e^{- x-a }, & x < 3 \\ \sqrt[5]{a + x^{1.5}}, & 3 \leq x < 10 \\ \ln^2(bx + 1), & x \geq 10 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} x < 3 \\ 3 \leq x < 10 \\ x \geq 10 \end{aligned}$
15	$y = \begin{cases} \sqrt{ a + x } + \sqrt[3]{ ax + 1}, & 0 \leq ax \leq 3 \\ e^{-x^2+a}, & ax < 0 \\ x^{\cos(x)} + \frac{b}{x^2 + a^2}, & ax \geq 3 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} 0 \leq ax \leq 3 \\ ax < 0 \\ ax \geq 3 \end{aligned}$
16	$y = \begin{cases} e^{-x^2-a^2}, & bx \leq 1 \\ \sqrt[3]{\frac{ ax + 1}{bx + 2,1}}, & 1 < bx < 5 \\ a \ln^2(bx), & bx \geq 5 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} bx \leq 1 \\ 1 < bx < 5 \\ bx \geq 5 \end{aligned}$
17	$y = \begin{cases} \sqrt[5]{\frac{x^2 + (ab)^2 + 1}{ ax + 1}}, & ax < 0 \\ \ln^2(ax + 1), & 0 \leq ax < 4 \\ \cos^2(a + \sqrt[3]{x^2}), & ax \geq 4 \end{cases}$	a, b - параметри; x, y - змінні	$\begin{aligned} ax < 0 \\ 0 \leq ax < 4 \\ ax \geq 4 \end{aligned}$
18	$y = \begin{cases} \ln^2 x - \frac{\pi}{6}, & x \geq 3 \\ \sqrt[3]{x} + e^{-x^2+a}, & 1 \leq x \leq 3 \\ \sin^3(b + ax), & x < 1 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} x \geq 3 \\ 1 \leq x \leq 3 \\ x < 1 \end{aligned}$
19	$y = \begin{cases} \sqrt[5]{ b + \sin^2(ax) + 1}, & x < -1 \\ 5 + \operatorname{arctg}\left(\sqrt[3]{ bx + 1}\right), & -1 \leq x \leq 2 \\ \frac{x}{eab}, & x > 2 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} x < -1 \\ -1 \leq x \leq 2 \\ x > 2 \end{aligned}$
20	$y = \begin{cases} \sqrt[3]{b^2 + \cos^3 ax}, & x \leq 1 \\ a^x + \ln^2 bx , & -1 \leq x < -0,5 \\ \operatorname{arctg} ax + \ln(2 + x), & x \geq -0,5 \end{cases}$	a, b - параметри; x, y - змінні	$\begin{aligned} x \leq 1 \\ -1 \leq x < -0,5 \\ x \geq -0,5 \end{aligned}$
21	$y = \begin{cases} \sqrt[5]{ b + \sin^2 ax + 1}, & x < -1 \\ 5 + \operatorname{arctg} \sqrt[3]{ bx + 1}, & -1 \leq x \leq 2 \\ e^{-\frac{x}{ab}}, & x > 2 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} x < -1 \\ -1 \leq x \leq 2 \\ x > 2 \end{aligned}$
22	$y = \begin{cases} \cos(ax) + \lg(ax + 2), & x > 3 \\ a\sqrt{b^2 + e^x}, & 1 \leq x \leq 3 \\ \frac{x^2}{2} + \frac{x^4}{4}, & x < 1 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} x > 3 \\ 1 \leq x \leq 3 \\ x < 1 \end{aligned}$
23	$y = \begin{cases} a + \ln^2(1 + e^{bx}), & x \geq 0 \\ \frac{\pi + \sin^2 ax}{\sqrt[3]{a^2 + bx + x^2 + 1}}, & -2 < x < 0 \\ \left(\frac{ab}{a^4 + x^2 + 1}\right), & x \leq -2 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$\begin{aligned} x \geq 0 \\ -2 < x < 0 \\ x \leq -2 \end{aligned}$

24	$y = \begin{cases} \sin^3 bx + \sqrt[3]{\cos^2 \pi x}, & x \leq 1 \\ \lg^2(ax + b + 1) - \sqrt{ x }, & 1 < x < 8 \\ e^{\frac{ax}{-a^2+b^2+1}}, & x \geq 8 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x \leq 1$ $1 < x < 8$ $x \geq 8$
25	$y = \begin{cases} \frac{ax+3}{\pi + \cos bx}, & x \leq 2 \\ e^{-x^2+ x }, & 2 < x \leq 4 \\ \lg^2\left(\sin ax + \sqrt[3]{x^2 + a^2 + 1}\right), & x > 4 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x \leq 2$ $2 < x \leq 4$ $x > 4$
26	$y = \begin{cases} \sin^2 \sqrt{x^2 + a^2 + 1}, & x < 0 \\ \frac{x^2}{2} + bv , & x > 2 \\ e^{-4ax+b}, & 0 < x < 2 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x < 0$ $x > 2$ $0 < x < 2$
27	$y = \begin{cases} \frac{\cos^2 x}{x^2 + a^4 x + b + 1}, & x \geq 1 \\ \sqrt{\lg^2(bx + 1.5 + 2)}, & x \leq -1 \\ \sin x + e^{ax}, & -1 < x < 1 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x \geq 1$ $x \leq -1$ $-1 < x < 1$
28	$y = \begin{cases} \sqrt{ x + a^2}, & x \leq -2 \\ \lg \left \frac{\sqrt[3]{ ax + b + \pi}}{1 + \sin^2(\pi x)} \right , & x > 1 \\ x^5 + \operatorname{arctg}(bx), & -2 < x \leq 1 \end{cases}$	a, b - параметри; x, y - змінні	$x \leq -2$ $x > 1$ $-2 < x \leq 1$
29	$y = \begin{cases} \arcsin(ax + b), & x > 0 \\ \ln(2 + \cos^2(bx)), & x \leq -2 \\ \sin^3(ax + b)^2, & -2 < x \leq 0 \end{cases}$	a, b - параметри; x, y - змінні	$x > 0$ $x \leq -2$ $-2 < x \leq 0$
30	$y = \begin{cases} \arccos(bx + a), & x \leq 3 \\ \frac{x+ab-3}{e^{\sqrt{ x^2+ax-b +1}}}, & x > 5 \\ \lg^3(\cos(ax) + b^2 + 2), & 3 < x \leq 5 \end{cases}$	e – стала (2.718); a, b - параметри; x, y - змінні	$x \leq 3$ $x > 5$ $3 < x \leq 5$

ЛАБОРАТОРНА РОБОТА №6: “ПРОГРАМУВАННЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ ОПЕРАТОРА IF НА ПРИКЛАДІ ВИЗНАЧЕННЯ НАЛЕЖНОСТІ ТОЧКИ ОБЛАСТІ”

МЕТА РОБОТИ

1. Вивчити оператор if мови C/C++.
2. Вивчити команду ? мови C/C++.
3. Ознайомитися з технологією створення розгалужених програм засобами мови C/C++.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою *lab_work_06_01* для розв’язування такої задачі. Дано координати точки x та y точки $M(x, y)$ координатної площини. Визначити, чи належить ця точка області, заштрихованій на малюнку (рис. 1).

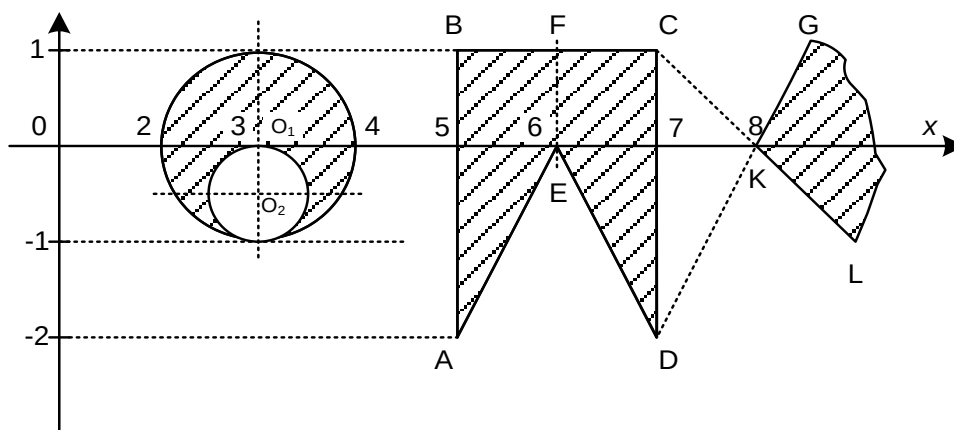


Рисунок 1. Область, про яку йдеться в умові задачі

Рівняння ліній, які обмежують область (їх рекомендується скласти самостійно).

Велике коло з центром в точці $O_1(3; 0)$: $(x-3)^2 + y^2 = 1$;

Мале коло з центром в точці $O_2(3; -0.5)$: $(x-3)^2 + (y+0.5)^2 = 0.25$;

Пряма AB:	$x=5$.	Пряма BF:	$y=1$.
Пряма FE:	$x=6$.	Пряма AE:	$y=2x-12$.
Пряма EF:	$x=6$.	Пряма FC:	$y=1$.
Пряма CD:	$x=7$.	Пряма ED:	$y=-2x+12$.
Пряма KG:	$y=2x-16$.	Пряма KL:	$y=-x+8$.

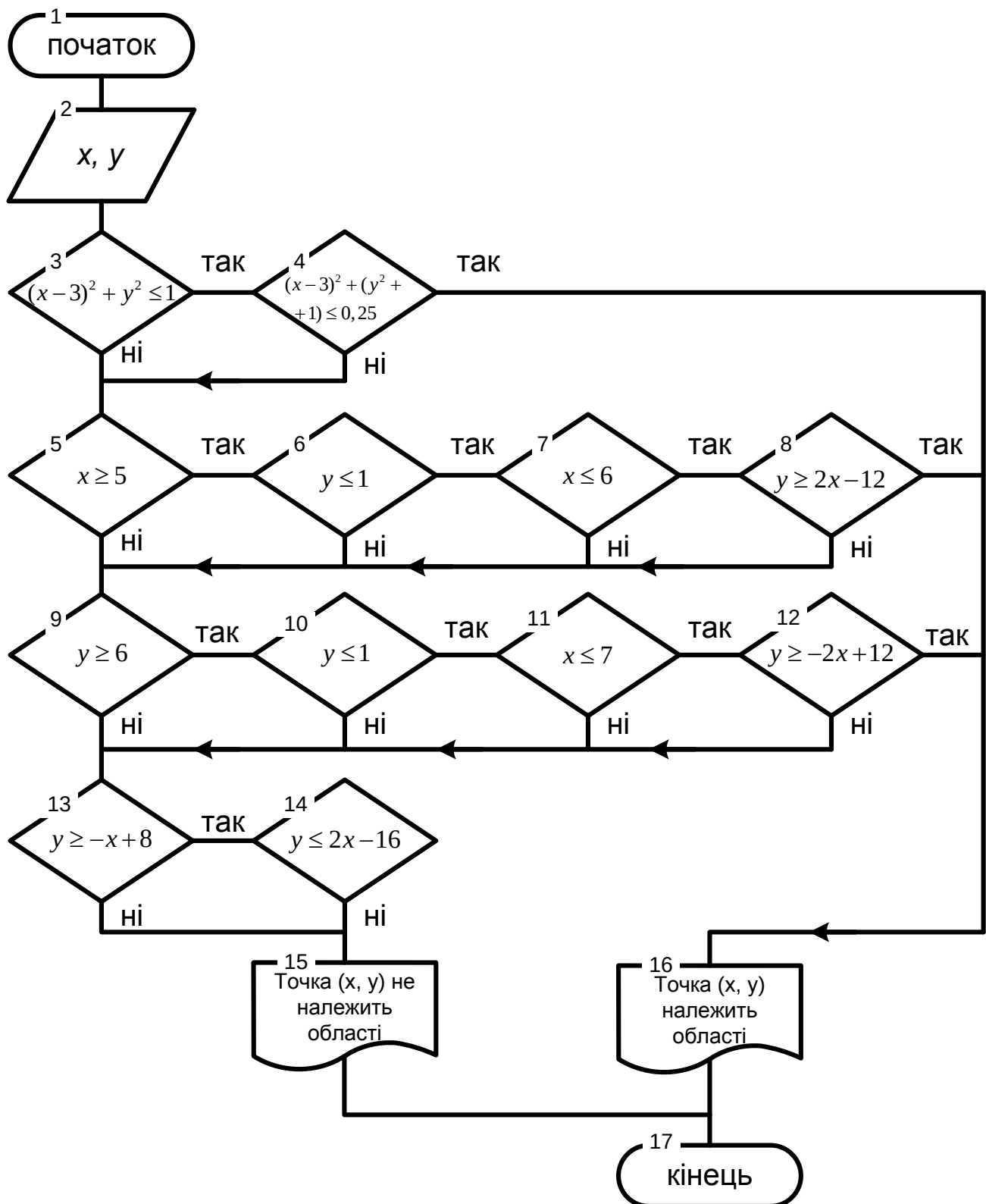


Рисунок. 2. Алгоритм розв'язування задачі про належність точки $M(x, y)$ заштрихованій області

```

/*****
/*      lab_work_06_01.                                */
/*      Необхідно визначити, чи належить точка M(x,y)  з      */
/* заданими координатами x та y (які мають дійсний тип) області, */
/* яка заштрихована на малюнку.                                */
/* Дано:  x, y - значення координат точки;                    */
/*      Необхідно:                                             */

```

```

/* надрукувати повідомлення: Точка М(х,у) належить          */
/* заштрихованій області, якщо це так;                      */
/* або повідомлення: Точка М(х,у) не належить                */
/* заштрихованій області, якщо це не так;                    */
/* Реалізувати програму мовою С.                              */
/*****/
#include <stdio.h>
#include <conio.h>
#include <math.h>

//Оголошення змінних
float x;    // Абсциса точки М
float y;    // Ордината точки М
int main() {
    clrscr();
    // Введення значень абсциси та ординати точки:
    printf ("\n Введіть значення абсциси та ординати: ");
    scanf ("%f%f", &x, &y);
    //Визначення чи належить точка М(х,у) заштрихованій
    //на малюнку області
    if (((x-3)*(x-3)+y*y <=1) && ((x-3)*(x-3)+(y+0.5)*(y+0.5)
                                     >=0.25)) ||
        ((x>=5) && (y<=1) && (x<=6) && (y>=2*x-12)) ||
        ((x>=6) && (y<=1) && (x<=7) && (y>=-2*x+12)) ||
        ((y>=-x+8) && (y<=2*x-16)))
        printf("\nx=%f y=%f Точка М(х,у) належить заштрихованій
області. ", x,y);
    else
        printf("\nx=%f y=%f Точка М(х,у) не належить
заштрихованій області. ", x,y);
    getch();
    return 0;
}

```

Результати тестування програми:

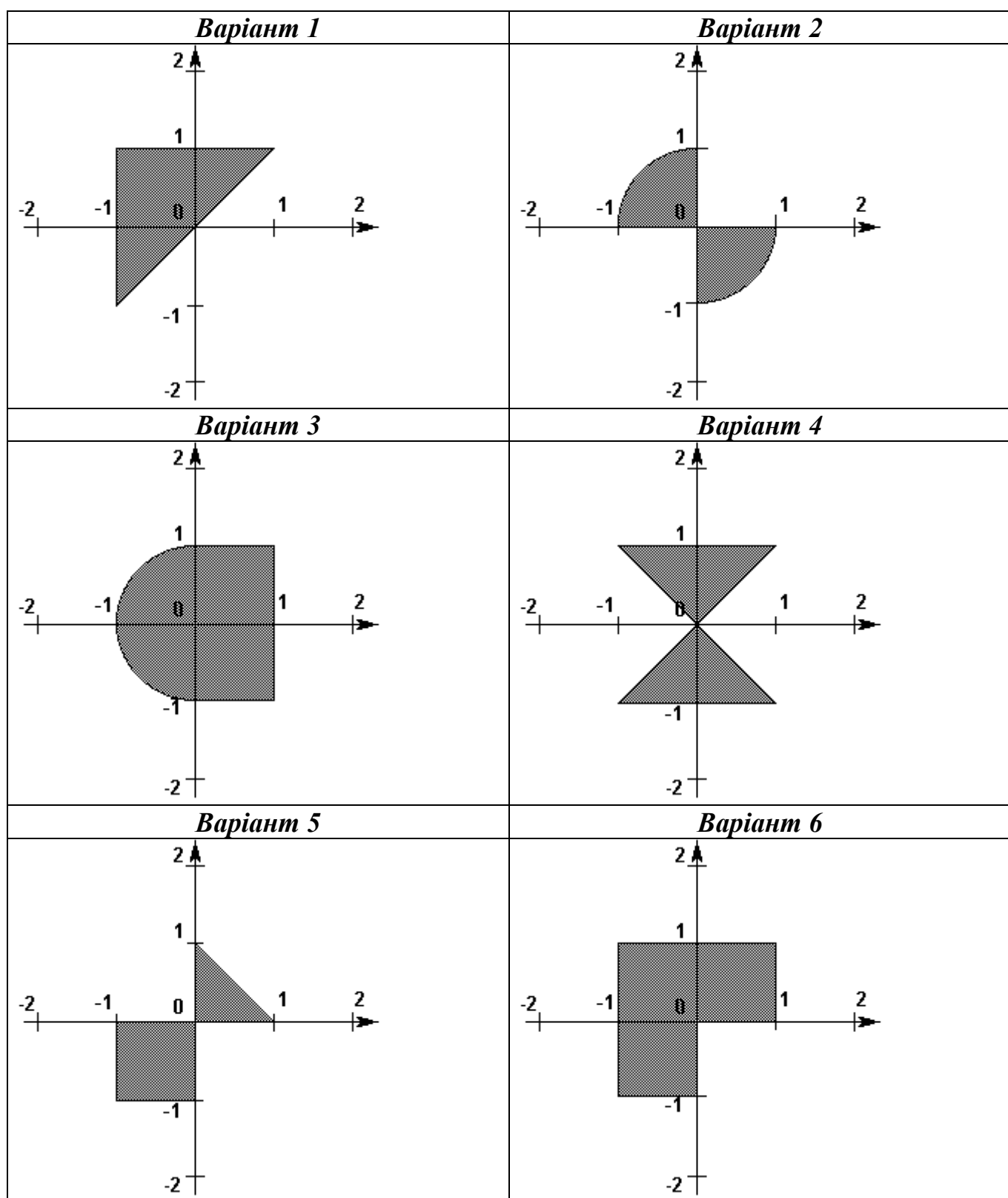
```

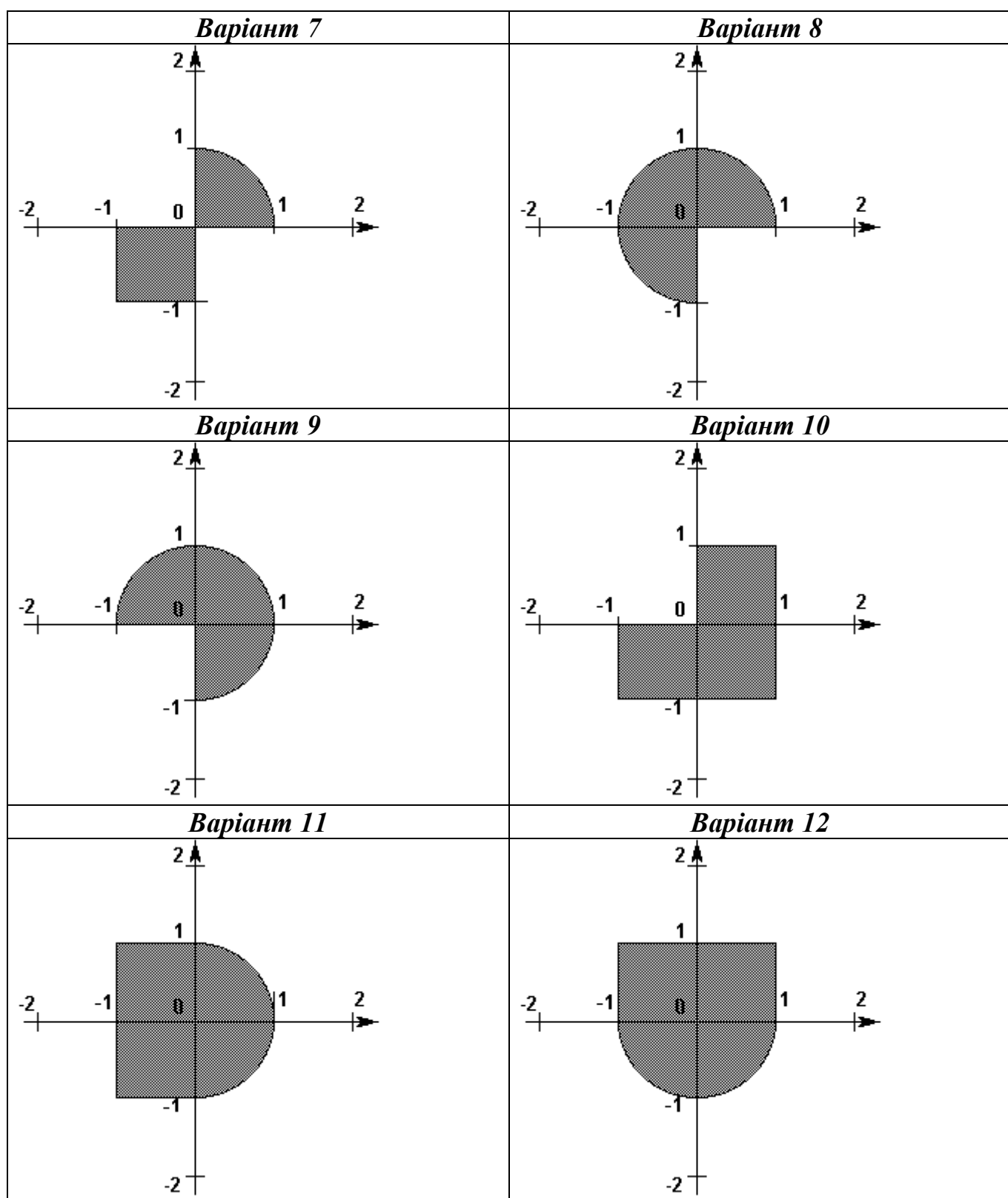
x=3.0 y=0.5 Точка М(х,у) належить заштрихованій області
x=5.5 y=0.5 Точка М(х,у) належить заштрихованій області
x=6.5 y=0.5 Точка М(х,у) належить заштрихованій області
x=9.0 y=0.5 Точка М(х,у) належить заштрихованій області
x=1.0 y=1.5 Точка М(х,у) не належить заштрихованій області
x=9.0 y=9.5 Точка М(х,у) не належить заштрихованій області

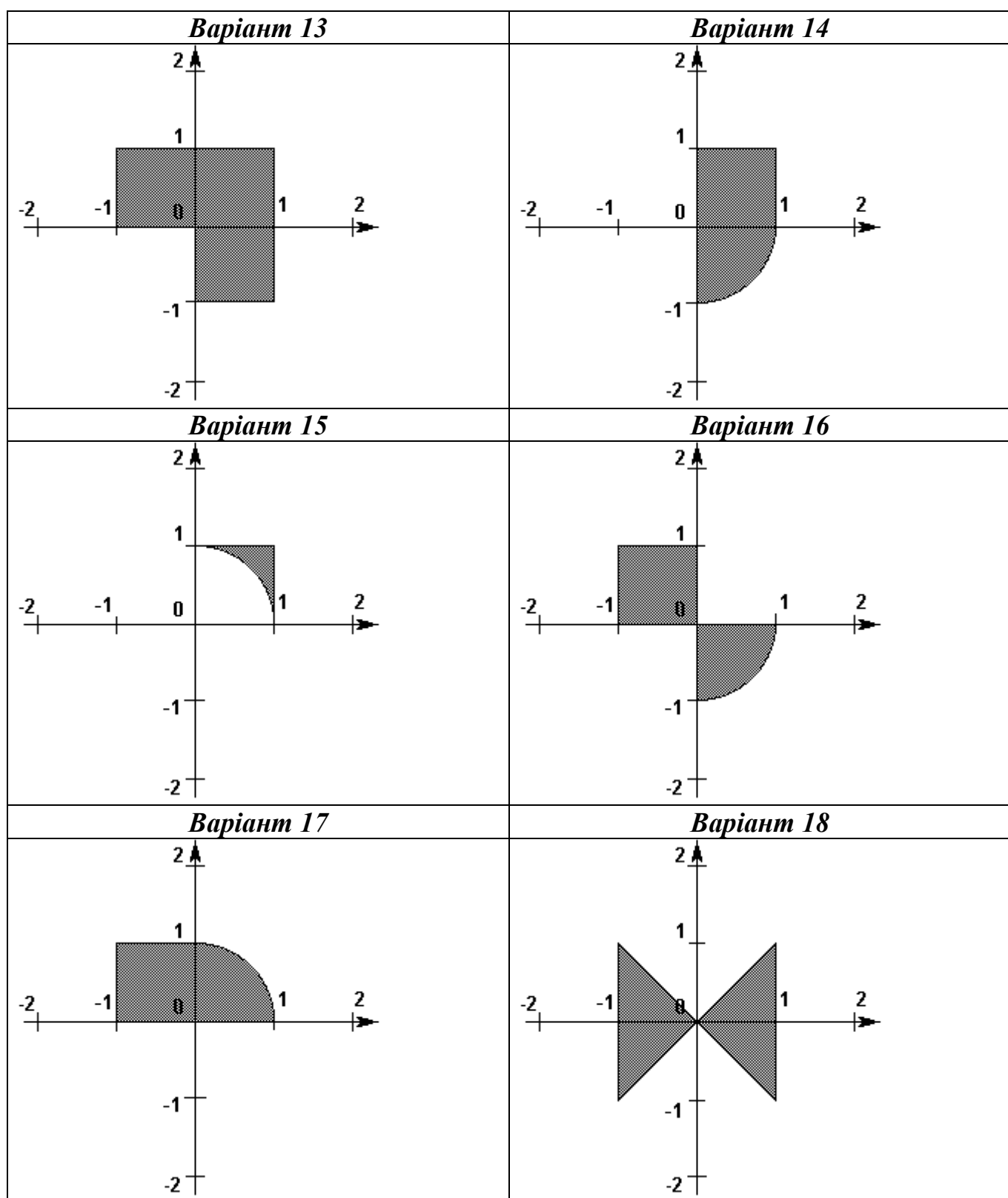
```

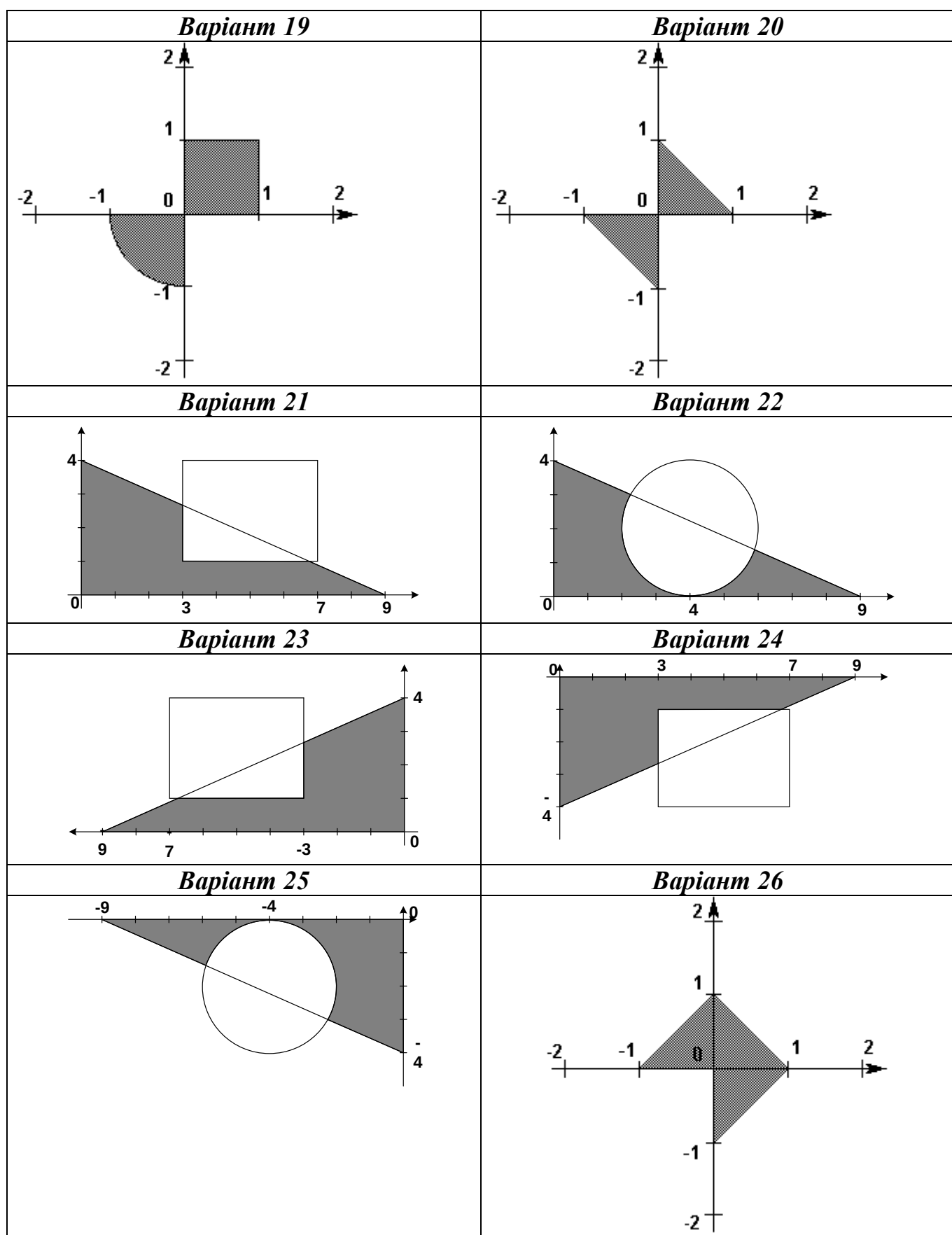
ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

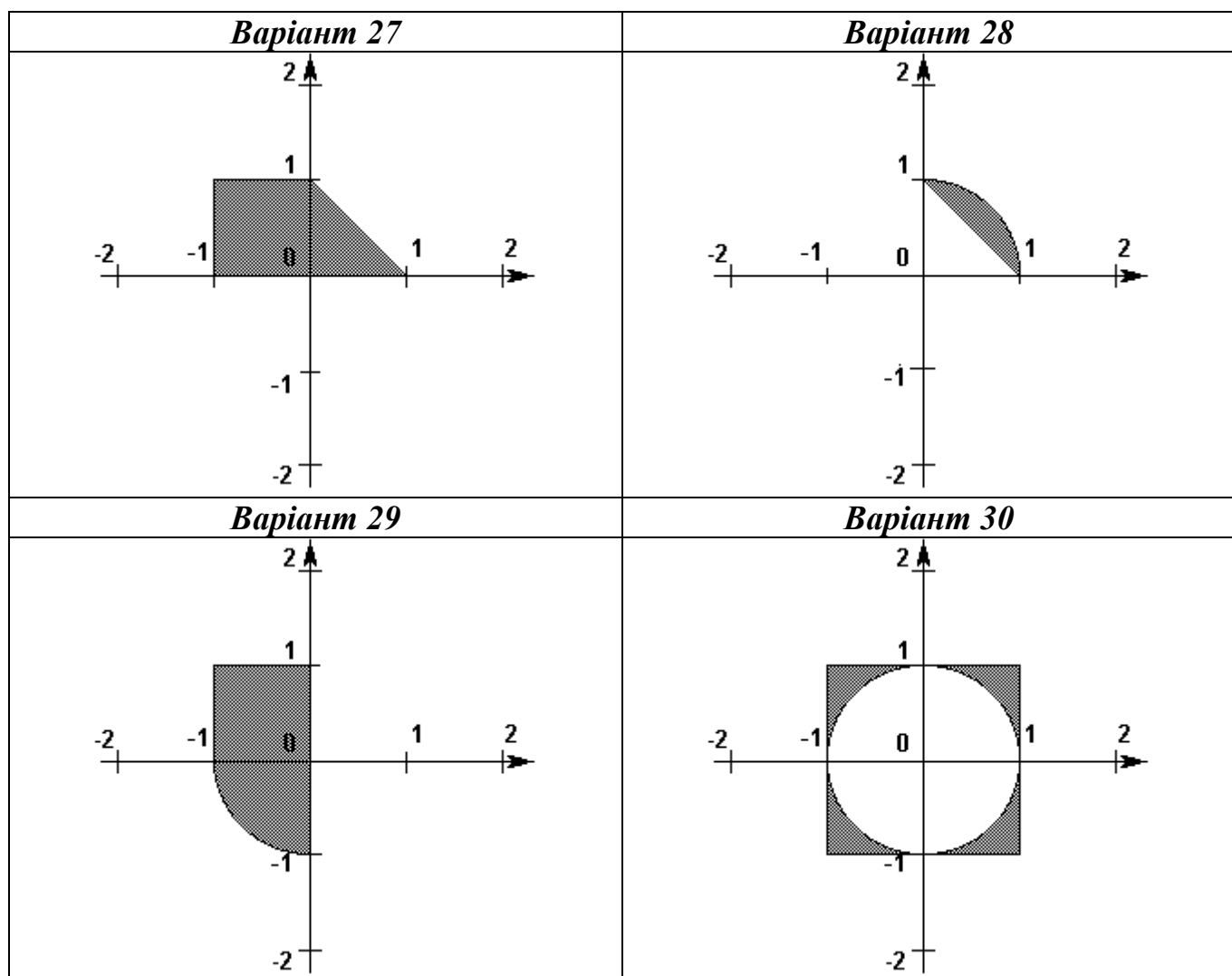
Скласти, відлагодити та протестувати розгалужену програму для визначення належності точки М зі заданими координатами x та y за допомогою умовного оператора `if`.











ЛАБОРАТОРНА РОБОТА №7: “ПРОГРАМУВАННЯ РОЗГАЛУЖЕНИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ ОПЕРАТОРА ВИБОРУ SWITCH”

МЕТА РОБОТИ

1. Вивчити оператор switch мови C/C++.
2. Ознайомитися з технологією створення розгалужених програм за допомогою оператора switch.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму визначення положення точки $M(x, y)$ з заданими координатами на координатній площині за допомогою оператора switch і зберегти її під назвою `lab_work_07_01`.

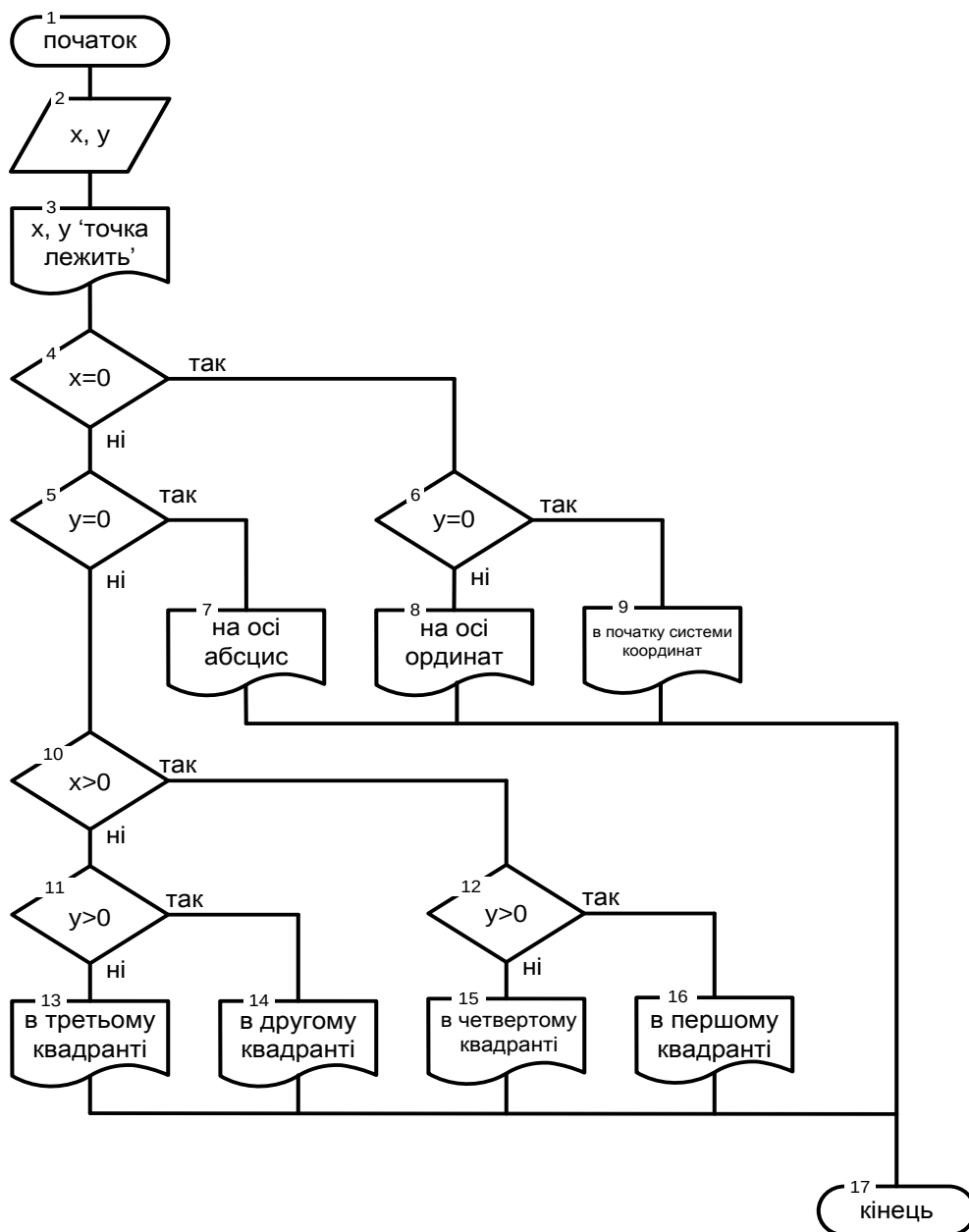


Рис. 3. Алгоритм визначення положення точки на координатній площині

```

/*****
/* lab_work_07_01
/* Визначити номер квадранта, в який попадає точка M(x,y) зі
/* заданими координатами x,y,z засобами мови C,
/* використовуючи switch оператор.
/* Дано: x, y - значення координат точки M - довільні
/* дійсні числа;
/* Необхідно:
/* Визначити квадранта октанта, в який попадає точка M(x,y,z).
/* Розробити програму засобами мови C, використовуючи switch
/* оператор.
*****/

#include <stdio.h>
#include <conio.h>
#include <math.h>

//Оголошення змінних:
float x; // абсциса точки M
float y; // ордината точки M

int flag; //змінна flag значення якої використовується для
//визначення квадранта, до якого належить точка M(x,y)
int main() {
    clrscr();

    //Введення значень координат точки M(x,y):
    printf ("\n Ввести значення координат x,y точки M: ");
    scanf("%f%f",&x, &y);

    // Визначення осі або номера квадранта,
    // в який попадає точка M(x,y,z) зі заданими координатами
    if ((x==0) && (y == 0)) flag=1;
    if ((x==0) && (y != 0)) flag=2;
    if ((y==0) && (x != 0)) flag=3;
    if ((x>0) && (y>0)) flag=4;
    if ((x<0) && (y>0)) flag=5;
    if ((x<0) && (y<0)) flag=6;
    if ((x>0) && (y<0)) flag=7;

    printf("\n x=%6.2f y=%6.2f.", x,y);

    //Друк назви осі чи номера квадранта ,в який попадає точка
    //M(x,y) зі заданими координатами
    switch (flag) {
        case 1: printf("\n Точка M(x,y) лежить в початку
                        координат.");
                break;
        case 2: printf("\n Точка M(x,y) лежить на осі 0x.");
                break;
        case 3: printf("\n Точка M(x,y) лежить на осі 0y.");

        case 4: printf("\n Точка M(x,y) належить 1 квадранту.");
    }
}

```

```

        break;
    case 5: printf("\n Точка  М(х,у) належить  2 квадранту.");
        break;
    case 6: printf("\n Точка  М(х,у) належить  3 квадранту.");
        break;
    case 7: printf("\n Точка  М(х,у) належить  4 квадранту.");
        break;
}
getch();
return 0;
}

```

Результати тестування програми:

Ввести значення координат x, y точки M : 0 0
 $x = 0.0$ $y = 0.0$ Точка $M(x, y)$ лежить в початку координат.
Ввести значення координат x, y точки M : 1 0
 $x = 1.0$ $y = 0.0$ Точка $M(x, y)$ лежить на осі Ox .
Ввести значення координат x, y точки M : 0 1
 $x = 0.0$ $y = 1.0$ Точка $M(x, y)$ лежить на осі Oy .
Ввести значення координат x, y точки M : 1 1
 $x = 1.0$ $y = 1.0$ Точка $M(x, y)$ належить 1 квадранту.
Ввести значення координат x, y точки M : -1 1
 $x = -1.0$ $y = 1.0$ Точка $M(x, y)$ належить 2 квадранту.
Ввести значення координат x, y точки M : -1 -1
 $x = -1.0$ $y = -1.0$ Точка $M(x, y)$ належить 3 квадранту.
Ввести значення координат x, y точки M : 1 -1
 $x = 1.0$ $y = -1.0$ Точка $M(x, y)$ належить 4 квадранту.

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати дві розгалужені програми розв'язування завдань, заданих у лабораторних роботах №5 та №6 за допомогою умовного оператора `switch`.

ЛАБОРАТОРНА РОБОТА №8: “ПРОГРАМУВАННЯ ЦИКЛІЧНИХ ПРОЦЕСІВ МОВОЮ C/C++ ЗА ДОПОМОГОЮ ОПЕРАТОРА FOR”

МЕТА РОБОТИ

1. Вивчити оператор циклу `<for>` мови C/C++.
2. Ознайомитися з технологією створення циклічних програм на базі оператора `<for>`.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_08_01` для розв’язування наступної задачі. Дано: функцію $y=2\sin(3x)+3\cos(5x)$, аргумент x якої змінюється на інтервалі $[2; 4]$ з кроком $h=0,2$. Необхідно обчислити та вивести на друк: всі значення аргумента та функції; суму, кількість та середнє арифметичне тих значень функції, які задовільняють нерівності $1 \leq y \leq 3$. Якщо таких значень не виявиться, вивести на друк відповідне повідомлення. Циклічні процеси реалізувати командою `for` мовою C.

```
/* **** */
/* lab_work_08_01. */
/* Табулювання функції на інтервалі засобами мови C. */
/* Обчислити всі значення функції */
/* y=2*sin(3*x)+3*cos(5*x), аргумент x якої змінюється */
/* на інтервалі [2;4] з кроком h=0,2. Обчислити також */
/* суму, кількість та середнє арифметичне тих значень */
/* функції, які задовільняють нерівність 1<=y<=3. */
/* Якщо таких значень не виявиться, вивести */
/* відповідне повідомлення. */
/* Циклічні процеси реалізувати командою for. */
/* **** */
#include <stdio.h>
#include <conio.h>
#include <math.h>

void main() {
    clrscr();

    float x; // Аргумент функції
    float y; // Значення функції
    float s=0; // Сума значень функції з інтервалу [1;3]
    float sa; // Середнє арифметичне значень функції
                // з інтервалу [1;3]
    int k=0; // Кількість значень функції з інтервалу [1;3]

    // Табулювання функції на заданому інтервалі
    // з одночасним обчисленням кількості k та суми s:
    printf("\n Результати табулювання функції:\n");
    x=2;
    for (x=2; x<4.1; x+=0.2) {
        y=2*sin(3*x)+3*cos(5*x);
        printf(" x=%+4.2f y=%+4.2f\n", x, y);
        if ((y>=1) && (y<=3))
```

```

    {
        k=k+1;
        s=s+y;
    }
}

printf("\n  Обробка результатів табулювання функції:\n");
if (k>0) {
    sa=s/k;
    printf( "\n  Кількість значень функції
            з інтервалу [1;3] k=%3d",k);
    printf( "\n  Сума значень функції
            з інтервалу [1;3] s=%6.2f",s);
    printf( "\n  Середнє арифметичне значень функції
            з інтервалу [1;3] sa=%6.2f\n",sa);
}
else
    printf( "\n  k=%3d Значень функції
            з інтервалу [1;3] немає \n",k);

getch();
}

```

Результати тестування програми:

```

x=+2.00 y=-3.08
x=+2.20 y=+0.64
x=+2.40 y=+4.12
x=+2.60 y=+4.72
x=+2.80 y=+2.12
x=+3.00 y=-1.45
x=+3.20 y=-3.22
x=+3.40 y=-2.23
x=+3.60 y=+0.02
x=+3.80 y=+1.13
x=+4.00 y=+0.15

```

Обробка результатів табулювання функції:

```

Кількість значень функції з інтервалу [1;3] k=  2
Сума значень функції з інтервалу [1;3] s=  3.25
Середнє арифметичне значень функції з інтервалу [1;3] sa=  1.62

```

2. Ввести, відлагодити та протестувати програму і зберегти її під назвою lab_work_08_02 для розв'язування наступної задачі. Дано: функцію $y=2*\sin(3x)+3*\cos(5x)$, аргумент x якої змінюється на інтервалі $[2;4]$ з кроком $h=0,2$. Необхідно: обчислити та вивести на друк всі значення аргумента та функції; знайти мінімальне значення $\min$ функції $y=f(x)$ та відповідне йому значення аргумента $x_{\min}$; знайти максимальне значення $\max$ функції $y=f(x)$ та відповідне значення аргумента $x_{\max}$. Циклічні процеси реалізувати командою `for` мовою C. Для обчислення значень функції створити функцію користувача.

```

/*****
/*  lab_work_08_02.
/*  Пошук мінімального min та максимального max значень функції
/*   $y=f(x)$ , заданої на інтервалі[a;b]
/*  засобами мови C за допомогою оператора for.
/*  Дано функцію  $y=2*\sin(3*x)+3*\cos(5*x)$ , аргумент x якої
/*  змінюється на інтервалі [2; 4] з кроком  $h = 0,2$ .
/*  Необхідно: обчислити і вивести на друк у вигляді таблиці
/*  всі значення функції;
/*  знайти мінімальне значення min функції  $y=f(x)$ 
/*  та відповідне йому значення аргумента xmin;
/*  знайти максимальне значення max функції  $y=f(x)$ 
/*  та відповідне значення аргумента xmax;
/*  Створити програму засобами мови C,
*/  використовуючи оператор for.
/*  Для обчислення значень функції створити функцію користувача.
*****/

#include <stdio.h>
#include <conio.h>
#include <math.h>

//Declaration of variables:
float x;    // значення аргумента функції
float xf;   // значення аргумента у підпрограми-функції
float y;    // значення функції
float min, max;    // мінімальне та максимальне значення
функції
float xmin, xmax; // значення аргументів функції, у яких вона
досягає значень min та max

float  f(float xf); //Прототип підпрограми-функції

void main()    {
    clrscr();
    xmin=2;    //Початкове значення xmin
    xmax=4;    // Початкове значення xmax
    min=f(2);  // Початкове значення min
    max=f(2);  // Початкове значення max
    // Табулювання функції  $y = 2 * \sin (3 * x) + 3 * \cos (5 * x)$ 
    // знаходження величин min, xmin, max, xmax:
    printf("\n Результати табулювання функції
            $y=2*\sin(3*x)+3*\cos (5 * x):\n");$ 
    x=2;
    for (x=2;   x<4.1;   x+=0.1)    {
        y=f(x);
        printf("  x=%+4.2f y=%+4.2f\n",x,y);
        if (y<min) {
            min=y;
            xmin=x;
        }
        if (y>max) {
            max=y;

```

```

        xmax=x;
    }
}
printf("\n Результати аналізу значень функції: \n");
printf( "\n Мінімальне значення функції min=%6.2f   в точці
        x=%6.2f.",min, xmin);
printf( "\n Максимальне значення функції max=%6.2f   в точці
        x=%6.2f.",max, xmax);

    getch();
}

//-----
// Ця підпрограма-функція обчислює значення
// функції 2*sin(3*xf)+3*cos(5*xf),
//яке передається у головну програму через ім'я функції f

float  f(float xf) {
    return  2 * sin(3 *xf) + 3 * cos(5 * xf);
}

```

Результати тестування програми:

```

x=+2.00 y=-3.08
x=+2.20 y=+0.64
x=+2.40 y=+4.12
x=+2.60 y=+4.72
x=+2.80 y=+2.12
x=+3.00 y=-1.45
x=+3.20 y=-3.22
x=+3.40 y=-2.23
x=+3.60 y=+0.02
x=+3.80 y=+1.13
x=+4.00 y=+0.15

```

Результати аналізу значень функції:
 Мінімальне значення функції min=-3.22 в точці x=+3.20.
 Максимальне значення функції max=+4.72 в точці x=+2.60

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати циклічну програму розв'язування задачі табулювання функції $y=f(x)$ на інтервалі $[x_{\text{поч.}} \ x_{\text{кін}}]$ з кроком h за допомогою оператора `for`. У програмі передбачати обчислення величин, вказаних у варіантах завдань.

№	$y=f(x)$	$x_{\text{поч}}$	$x_{\text{кін}}$	h	a	b	Величини, які необхідно обчислити
1	$y = \frac{e^{\sin x} + \sqrt[4]{a+x}}{\ln^3 bx}$	5.5	10.5	0.5	17.3	0.36	Середнє арифметичне $y > 5$

№	$y=f(x)$	$x_{\text{поч}}$	$x_{\text{кін}}$	h	a	b	Величини, які необхідно обчислити
2	$y = \frac{\ln^4 bx + 0.85}{\sqrt[3]{a + bx^3}}$	0.4	6.8	0.8	46	1.85	Суму і кількість $y > 1$
3	$y = \frac{\sqrt[4]{1 + \sqrt{ax + b}}}{\sin^2 bx + a}$	4.3	13.9	1.2	1.35	8.4	Суму $1 < y < 3$
4	$y = \frac{tg^2(ax) + \sqrt{\ln x}}{e^{-bx}}$	1.3	6.1	0.6	1.8	0.56	Середнє арифметичне $y > 1$
5	$y = \frac{\sqrt[3]{a^3 + x^3}}{tg^3 bx + 1.6}$	0.2	1.6	0.2	1.25	0.86	Суму $0 < y < 1$
6	$y = \frac{\sqrt[3]{ax} + b}{0.25 \ln^2 ax}$	10.5	28.5	2	0.3	9.5	Добуток і кількість $y < 10$
7	$y = \frac{\ln^2 (a^3 + x^3)}{\sqrt{a^3 + x^3} + \sqrt[3]{b}}$	8.2	98.2	10	43	205	Суму і кількість $y > 0.2$
8	$y = \frac{1 + \cos^2 (a^2 + x^2)}{x^3 + \sqrt[3]{bx}}$	0.5	1.9	0.2	0.84	0.63	Середнє геометричне $y > 1$
9	$y = \frac{a^x + e^{-bx}}{\sin^2 bx + 1.24}$	0.3	1.3	0.1	0.5	0.16	Добуток і суму $y > 1$
10	$y = \sqrt{\frac{ bx }{b^2} \arctg \frac{b^2}{a^2 + x^2}}$	-10	-1	1	2.8	1.5	Середнє арифметичне $y > 0.5$
11	$y = \frac{(x^2 + 1) - \frac{1}{\sin bx}}{\sqrt[3]{\frac{x}{a}} - 0.39}$	0.2	1.6	0.1	0.36	0.74	Кількість $y > 0.5$ і добуток $y > 1$
12	$y = \frac{e^{x^2+1}}{\sqrt[5]{ x - a } + \ln^2 bx}$	1.2	3	0.2	4.8	6.8	Кількість $y > 10$ і суму $y < 10$
13	$y = \frac{e^{\sin^2 ax} + \arctg bx}{\sqrt[3]{(x + b)^2}}$	0.5	14.5	2	0.45	8.8	Кількість $y < 0.5$ і кількість $y > 0.5$
14	$y = \frac{\sin^2 ax + \sqrt[3]{ x - b }}{ x - b ^3}$	16	22	0.5	0.28	19.3	Суму і кількість $y > 2$

№	$y=f(x)$	$x_{\text{поч}}$	$x_{\text{кін}}$	h	a	b	Величини, які необхідно обчислити
15	$y = \frac{x^{\frac{a}{b}} - \sqrt[3]{\frac{x+b}{a}}}{1.1 + \cos^2 ax}$	6.8	20.8	1	3.5	6.4	Добуток і кількість $y < 10$
16	$y = \sqrt[5]{\frac{x^2}{4a^2 + 0.6}} \ln^2(b + x)$	10	20	1	2.4	16.8	Суму і кількість $y > 20$
17	$y = \frac{\cos^2 ax + e^{-ax}}{\arctg(\sqrt{b} + \sqrt{x})}$	0.6	0.8	0.02	1.3	0.75	Кількості $y < 2.75$ і $y > 2.75$
18	$y = \frac{a - e^{bx}}{(x - a)^2 + \ln^2(x - a)}$	2	8.5	0.5	4.38	0.24	Середнє арифметичне всіх y
19	$y = \frac{\sqrt[5]{a + x} - b}{1.6 \cos^{3.2} ax}$	0.3	3.3	0.3	1.85	2.64	Середнє геометричне $y > 0$
20	$y = \frac{e^{a+x}}{\sqrt[4]{bx} + \ln^2 ax}$	0.8	2.4	0.2	1.38	9.6	Середнє арифметичне $y > 5$
21	$y = \frac{\sqrt[4]{a + x} + \left(\frac{x}{b}\right)^{1.3}}{0.8 \operatorname{tg}^3 bx}$	8.5	20	0.5	17.6	0.14	Кількості $y > 0$ і $y < 0$
22	$y = \frac{\cos^2 x - a }{\sqrt[3]{ bx } + e^{x-a}}$	-5	5	1	1.5	14.8	Суму і кількість $y > 0.1$
23	$y = \frac{1 + \cos^2(a + bx)}{0.5 \sqrt[3]{a + bx}}$	2.5	16.5	2	33.6	6.45	Середнє арифметичне $y > 0.5$
24	$y = \frac{e^{\sin(x+b)} + a^3}{\ln^2(x + b)}$	0.25	7	0.5	0.86	2.8	Кількості $y < 0.5$ і $y > 0.5$
25	$y = \frac{\sqrt[3]{a + e^{bx}}}{a + \frac{\ln^2 x}{bx}}$	6.8	10	0.2	2.3	0.28	Суму $y > 0.5$ і кількість $y > 0.7$
26	$y = \frac{0.25 + \cos^2 ax}{\sqrt[4]{b^3 + x^3}}$	1.3	4.3	0.3	0.43	2.24	Суму $y < 0.5$ і добуток $y > 0.5$
27	$y = \frac{\arctg(a^2 + x^2)}{\ln^3 bx + \sqrt{ax}}$	1.1	3.6	0.5	1.38	2.46	Середнє арифметичне $y > 0.3$

№	$y=f(x)$	$x_{\text{поч}}$	$x_{\text{кін}}$	h	a	b	Величини, які необхідно обчислити
28	$y = \frac{e^{-6x} + a^2}{\sqrt[3]{\left(\frac{a}{x}\right)^3 + \sqrt{b}}}$	0.1	0.4	0.05	0.15	2.3	Добуток $y > 0.4$ і суму $y > 0.5$
29	$y = \frac{\sqrt{\ln a + \sqrt[3]{b^2 + 13}}}{\sin^2 ax + 0.84}$	0.04	0.24	0.02	4.2	1.54	Середнє геометричне $y > 0$
30	$y = \frac{\ln^3(a + x) - 10.5}{\sqrt[4]{0.6 b^2 + \left(\frac{x}{a}\right)^2}}$	56	166	10	32	8.4	Суму і добуток $y > 2$

Середнє арифметичне: $S = \frac{\sum_{i=1}^n y_i}{n}$

Середнє геометричне: $P = \sqrt[k]{\prod_{i=1}^k y_i}$

Середнє гармонійне: $G = \frac{1}{\frac{1}{n} \sum_{i=1}^n \frac{1}{y_i}}$

ЛАБОРАТОРНА РОБОТА №9: “ПРОГРАМУВАННЯ ЦИКЛІЧНИХ ПРОЦЕСІВ МОВОЮ C/C++ ЗА ДОПОМОГОЮ ОПЕРАТОРА WHILE”

МЕТА РОБОТИ

1. Вивчити оператор циклу <while> мови C/C++.
2. Ознайомитися з технологією створення циклічних програм на базі оператора <while>.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_09_01` для розв’язування наступної задачі. Дано: функцію $y=2\sin(3x)+3\cos(5x)$, аргумент x якої змінюється на інтервалі $[2;4]$ з кроком $h=0.2$. Необхідно обчислити та вивести на друк: всі значення аргумента та функції; суму, кількість та середнє арифметичне тих значень функції, які задовільняють нерівності $1 \leq y \leq 3$. Якщо таких значень не виявиться, вивести на друк відповідне повідомлення. Циклічні процеси реалізувати командою `while` мовою C.

```
/* **** */
/* lab_work_09_01. */
/* Табулювання функції на інтервалі засобами мови C. */
/* Обчислити всі значення функції */
/*  $y=2\sin(3x)+3\cos(5x)$ , аргумент  $x$  якої змінюється */
/* на інтервалі  $[2;4]$  з кроком  $h=0,2$ . Обчислити також */
/* суму, кількість та середнє арифметичне тих значень */
/* функції, які задовільняють нерівність  $1 \leq y \leq 3$ . */
/* Якщо таких значень не виявиться, вивести */
/* відповідне повідомлення. */
/* Циклічні процеси реалізувати командою while. */
/* **** */
#include <stdio.h>
#include <conio.h>
#include <math.h>

void main() {
    clrscr();

    float x; // Аргумент функції
    float y; // Значення функції
    float s=0; // Сума значень функції з інтервалу [1;3]
    float sa; // Середнє арифметичне значень функції
                // з інтервалу [1;3]
    int k=0; // Кількість значень функції з інтервалу [1;3]

    // Табулювання функції на заданому інтервалі
    // з одночасним обчисленням кількості k та суми s:
    printf("\n Результати табулювання функції:\n");
    x=2;
    while (x<=4.1) {
        y=2*sin(3*x)+3*cos(5*x);
        printf(" x=%+4.2f y=%+4.2f\n", x, y);
        if ((y>=1) && (y<=3)){
```



```

        k=k+1;
        s=s+y;
    }
    x=x+h;
}

printf("\n  Обробка результатів табулювання функції:\n");
if (k>0) {
    sa=s/k;
    printf( "\n  Кількість значень функції
            з інтервалу [1;3] k=%3d",k);
    printf( "\n  Сума значень функції
            з інтервалу [1;3] s=%6.2f",s);
    printf( "\n  Середнє арифметичне значень функції
            з інтервалу [1;3] sa=%6.2f\n",sa);
}
else
    printf( "\n  k=%3d Значень функції
            з інтервалу [1;3] немає \n",k);

getch();
}

```

Результати тестування програми:

```

x=+2.00 y=-3.08
x=+2.20 y=+0.64
x=+2.40 y=+4.12
x=+2.60 y=+4.72
x=+2.80 y=+2.12
x=+3.00 y=-1.45
x=+3.20 y=-3.22
x=+3.40 y=-2.23
x=+3.60 y=+0.02
x=+3.80 y=+1.13
x=+4.00 y=+0.15

```

Обробка результатів табулювання функції:

```

Кількість значень функції з інтервалу [1;3] k=  2
Сума значень функції з інтервалу [1;3] s=  3.25
Середнє арифметичне значень функції з інтервалу [1;3] sa=  1.62

```

2. Ввести, відлагодити та протестувати програму і зберегти її під назвою *lab_work_09_02* для розв'язування наступної задачі. Дано: числову послідовність, загальний член якої дорівнює $(-1)^n (5n+7) / (n^2+3n+1)$. де параметр n змінюється від 1 до 10. Необхідно: обчислити та вивести на екран в табличній формі значення всіх членів заданої числової послідовності;; обчислити та вивести на екран значення суми всіх членів послідовності; обчислити та вивести на екран значення добутку всіх членів послідовності. Циклічні процеси реалізувати командою `while` мовою C. Для обчислення значень загального члена послідовності створити функцію користувача.

```

/*****
/*  lab_work_09_02.
/*  Обчислення суми та добутку числової послідовності
/*  засобами мови C з використанням оператора while.
/*  Дано числову послідовність, загальний член якої дорівнює:
/*       $((-1)^n (5n+7) / (n^2+3n+1))$ .
/*  де параметр n змінюється від 1 до 10.
/*      Необхідно:
/*  обчислити та вивести на екран в табличній формі значення
/*  всіх членів заданої числової послідовності;
/*  обчислити та вивести на екран значення суми всіх членів
/*  послідовності;
/*  обчислити та вивести на екран значення добутку всіх членів
/*  Створити програму засобами мови C, використовуючи
/*  оператор циклу while.
*****/

```

```

#include <stdio.h>
#include <conio.h>
#include <math.h>

```

```

//Declaration of variables:
float sign, signf; // знак члена числової послідовності:
// sign - у головній функції; signf - у підпрограми-функції
float n, nf; // кількість членів числової послідовності:
// n - у головній функції; nf - у підпрограми-функції

```

```

float a; // поточне значення члена числової послідовності
float sum, d; // значення суми та добутку членів числової
// послідовності

```

```

float f(int signf, int nf); // Прототип підпрограми-функції

```

```

void main() {
    clrscr();
    sum=0; //Початкове значення суми sum
    d=1; //Початкове значення добутку d
    // Табулювання членів числової послідовності:
    // та обчислення суми та добутку її членів:
    printf("\n Результати табулювання членів числової
послідовності:\n");
    n=1;
    sign=-1;
    while (n<10.5) {
        a=f(sign, n);
        printf(" n=%3.0f a=%+10.5f\n",n,a);
        sum=sum+a;
        d=d*a;
        n++;
        sign=-sign;
    }
    n--;
    printf("\n Результати аналізу даних розрахунку: \n");
}

```

```

printf( "\n Сума %3d членів числової послідовності дорівнює
sum=%10.5f.",n, sum);
printf( "\n Добуток %3d членів числової послідовності дорівнює
d=%10.5f.",n, sum);

getch();

}

// Ця функція користувача обчислює і повертає значення
// заданої функції 2*sin(3*xf)+3*cos(5*xf);
float f(int signf, int nf) {
    return sign*(5*nf+7)/(nf*nf+3*nf+1);
}

```

Результати тестування програми:

```

n = 1    a=  -2.40000
n = 2    a=  +1.54545
n = 3    a=  -1.15789
n = 4    a=  +0.93103
n = 5    a=  -0.78049
n = 6    a=  +0.67273
n = 7    a=  -0.59155
n = 8    a=  +0.52809
n = 9    a=  -0.47706
n = 10   a=  +0.43511

```

Результати аналізу даних розрахунку:

Сума 10 членів числової послідовності дорівнює sum=-1.29458

Добуток 10 членів числової послідовності дорівнює d=-0.13614

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати циклічну програму розв'язування задачі обчислення суми чи добутку членів числової послідовності за даним у таблиці виразом за допомогою оператора циклу while.

№ п/п	Вираз для суми чи добутку	№ п/п	Вираз для суми чи добутку
1	$S = \sum_{i=1}^{10} \frac{(-1)^i i^2}{i^2 + 5i + 6}$	2	$S = \prod_{i=1}^{14} \frac{(-1)^{i+4} (5i^2 + i)}{6i^2 + i + 3}$
3	$S = \prod_{i=1}^{10} \frac{(-1)^i (i^2 + 5i + 6)}{i^2 + 5i}$	4	$S = \sum_{i=1}^{15} \frac{(-1)^{i+5} (6i^2 + 1)}{7i^2 + 2i + 4}$

№ п/п	Вираз для суми чи добутку	№ п/п	Вираз для суми чи добутку
5	$S = \sum_{i=1}^{11} \frac{(-1)^{i+1}(i+2)}{i^2 + 2i + 5}$	6	$S = \prod_{i=1}^{17} \frac{(-1)^{i+7}(5i^2 + i)}{4i^2 + 5i + 4}$
7	$S = \prod_{i=1}^{11} \frac{(-1)^{i+1}(i^2 + 2i + 5)}{i^2 + 2i}$	8	$S = \sum_{i=1}^{18} \frac{(-1)^{i+8}(6i^2 + 7)}{5i^2 + 6i + 5}$
9	$S = \sum_{i=1}^{12} \frac{(-1)^{i+2}i^2 + 4}{i^2 + 8i + 15}$	10	$S = \prod_{i=1}^{10} \frac{(-1)^i(i^2 + 4i)}{i^2 + 4i + 5}$
11	$S = \prod_{i=1}^{12} \frac{(-1)^{i+2}(i+8)(i+9)}{(i+7)(i+3)}$	12	$S = \sum_{i=1}^{10} \frac{(-1)^i(i^2 + 3)}{3i^2 + 2i + 2}$
13	$S = \sum_{i=1}^{13} \frac{(-1)^{i+3}(i^2 + 9)}{i^2 + 9i + 1}$	14	$S = \prod_{i=1}^{11} \frac{(-1)^{i+1}(2i^2 + 3i)}{i^2 + 3i + 4}$
15	$S = \prod_{i=1}^{13} \frac{(-1)^{1+4}(i^2 + 3i)}{2i^2 + 3i + 4}$	16	$S = \sum_{i=1}^{11} \frac{(-1)^{i+1}(i+1)}{i^2 + 2i + 7}$
17	$S = \sum_{i=1}^{14} \frac{(-1)^{i+4}}{5i^2 + i + 2}$	18	$S = \prod_{i=1}^{12} \frac{(-1)^{i+2}(i^2 + 2i)}{i^2 + 2i + 3}$
19	$S = \sum_{i=1}^{13} \frac{(-1)^{i+3}(3i^2 + 4)}{2i^2 + 2i + 1}$	20	$S = \prod_{i=1}^{16} \frac{(-1)^{i+6}(3i^2 + i)}{2i^2 + 3i + 2}$
21	$S = \prod_{i=1}^{12} \frac{(-1)^{i+2}(i+2)}{i^2 + 5i + 3}$	22	$S = \sum_{i=1}^{17} \frac{(-1)^{i+7}(4i^2 + 1)}{3i^2 + 4i + 3}$
23	$S = \sum_{i=1}^{13} \frac{(-1)^{i+3}(i^2 + 3)}{i^2 + 7i + 4}$	24	$S = \sum_{i=1}^{14} \frac{(-1)^{i+4}(6i^2 + 1)}{5i^2 + 4}$

№ п/п	Вираз для суми чи добутку	№ п/п	Вираз для суми чи добутку
25	$S = \prod_{i=1}^{14} \frac{(-1)^{i+4}(i^2 + 7)}{2i^2 + 5i + 3}$	26	$S = \prod_{i=1}^{15} \frac{(-1)^{i+5}(5i^2 + 2)}{4i^2 + 3}$
27	$S = \prod_{i=1}^{15} \frac{(-1)^{i+5}(7i^2 + 2i)}{8i^2 + 5i + 3}$	28	$S = \sum_{i=1}^{16} \frac{(-1)^{i+6}(7i^2 + 4)}{5i^2 + 3}$
29	$S = \sum_{i=1}^{16} \frac{(-1)^{i+6}(8i^2 + 1)}{9i^2 + 3i + 1}$	30	$S = \prod_{i=1}^{15} \frac{(-1)^{i+5}(i + 3)}{7i^2 + 4}$

ЛАБОРАТОРНА РОБОТА №10: “ПРОГРАМУВАННЯ ЦИКЛІЧНИХ ПРОЦЕСІВ МОВОЮ C/C++ ЗА ДОПОМОГОЮ ОПЕРАТОРА DO...WHILE”

МЕТА РОБОТИ

1. Вивчити оператор циклу <do...while> мови C/C++.
2. Ознайомитися з технологією створення циклічних програм на базі оператора <do..while>.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою *lab_work_10_01* для розв’язування наступної задачі. Дано: функцію $y=2*\sin(3*x)+3*\cos(5*x)$, аргумент x якої змінюється на інтервалі $[2;4]$ з кроком $h=0.2$. Необхідно обчислити та вивести на друк: всі значення аргумента та функції; суму, кількість та середнє арифметичне тих значень функції, які задовільняють нерівності $1 \leq y \leq 3$. Якщо таких значень не виявиться, вивести на друк відповідне повідомлення. Циклічні процеси реалізувати командою `do ... while` мовою C.

```
/* **** */
/*      lab_work_10_01.                               */
/*  Табулювання функції на інтервалі засобами мови C.  */
/*  Обчислити всі значення функції                      */
/*   $y=2*\sin(3*x)+3*\cos(5*x)$ , аргумент  $x$  якої змінюється */
/*  на інтервалі  $[2;4]$  з кроком  $h=0,2$ . Обчислити також */
/*  суму, кількість та середнє арифметичне тих значень */
/*  функції, які задовільняють нерівність  $1 \leq y \leq 3$ . */
/*  Якщо таких значень не виявиться, вивести          */
/*  відповідне повідомлення.                          */
/*  Циклічні процеси реалізувати командою do ... while.*/
/* **** */
#include <stdio.h>
#include <conio.h>
#include <math.h>

void main() {
    clrscr();

    float x;    // Аргумент функції
    float y;    // Значення функції
    float s=0;  // Сума значень функції з інтервалу [1;3]
    float sa;   // Середнє арифметичне значень функції
                // з інтервалу [1;3]
    int k=0;    // Кількість значень функції з інтервалу [1;3]

    // Табулювання функції на заданому інтервалі
    // з одночасним обчисленням кількості k та суми s:
    printf("\n  Результати табулювання функції:\n");
    x=2;
    do
        y=2*sin(3*x)+3*cos(5*x);
        printf("  x=%+4.2f y=%+4.2f\n", x, y);
```

```

    if ((y>=1) && (y<=3)){
        k=k+1;
        s=s+y;
    }
    x=x+h;
while (x<=4.1);

printf("\n  Обробка результатів табулювання функції:\n");
if (k>0) {
    sa=s/k;
    printf( "\n  Кількість значень функції
            з інтервалу [1;3] k=%3d",k);
    printf( "\n  Сума значень функції
            з інтервалу [1;3] s=%6.2f",s);
    printf( "\n  Середнє арифметичне значень функції
            з інтервалу [1;3] sa=%6.2f\n",sa);
}
else
    printf( "\n  k=%3d  Значень функції
            з інтервалу [1;3] немає \n",k);

getch();
}

```

Результати тестування програми:

```

x=+2.00 y=-3.08
x=+2.20 y=+0.64
x=+2.40 y=+4.12
x=+2.60 y=+4.72
x=+2.80 y=+2.12
x=+3.00 y=-1.45
x=+3.20 y=-3.22
x=+3.40 y=-2.23
x=+3.60 y=+0.02
x=+3.80 y=+1.13
x=+4.00 y=+0.15

```

Обробка результатів табулювання функції:

```

Кількість значень функції з інтервалу [1;3] k=  2
Сума значень функції з інтервалу [1;3] s=  3.25
Середнє арифметичне значень функції з інтервалу [1;3] sa=  1.62

```

2. Ввести, відлагодити та протестувати програму і зберегти її під назвою *lab_work_10_02* для розв'язування наступної задачі. Дано: числову послідовність, загальний член якої дорівнює:

$$\frac{(\sin x + \sin x \cdot \sin 2x + \sin x \cdot \sin 2x \cdot \sin 3x + \dots + \sin x \cdot \sin 2x \cdot \sin 3x \cdot \dots \cdot \sin kx)}{(\cos x + \cos x \cdot \cos 2x + \cos x \cdot \cos 2x \cdot \cos 3x + \dots + \cos x \cdot \cos 2x \cdot \cos 3x \cdot \dots \cdot \cos kx)},$$

де параметр k змінюється від 1 до 10. Необхідно: обчислити та вивести на екран в табличній формі значення всіх членів заданої числової послідовності; обчислити та вивести на екран значення суми всіх членів послідовності; обчислити та вивести на екран значення добутку всіх членів послідовності. Циклічні процеси реалізувати командою `do ...`

while мовою С. Для обчислення значень загального члена послідовності створити функцію користувача.

```
/* **** */
/* lab_work_10_02. */
/* Обчислення суми та добутку числової послідовності */
/* засобами мови С з використанням оператора do ... while. */
/* Дано числову послідовність, загальний член якої дорівнює: */
/*  $(\sin x + \sin x \sin 2x + \sin x \sin 2x \sin 3x + \dots + \sin x \sin 2x \sin 3x \dots \sin kx)$  */
/*  $(\cos x + \cos x \cos 2x + \cos x \cos 2x \cos 3x + \dots + \cos x \cos 2x \cos 3x \dots \cos kx)$ ; */
/* де параметр k змінюється від 1 до 10. */
/* Необхідно: */
/* обчислити та вивести на екран в табличній формі значення */
/* всіх членів заданої числової послідовності; */
/* обчислити та вивести на екран значення суми всіх членів */
/* послідовності; */
/* обчислити та вивести на екран значення добутку всіх членів */
/* Створити програму засобами мови С, використовуючи */
/* оператор циклу do ... while. */
/* **** */

#include <stdio.h>
#include <conio.h>
#include <math.h>

//Оголошення змінних:
float k; // кількість членів послідовності, сума яких обчислюється
float n; // кількість членів послідовності, добуток яких
обчислюється

float sum1; // значення суми членів послідовності у чисельнику
float sum2; // значення суми членів послідовності у знаменнику
float a; // значення дробу

float d1; // значення добутку членів послідовності
// у чисельнику дробу
float d2; // значення добутку членів послідовності у
// знаменнику дробу

float x; // задане значення аргумента
void main() {
    clrscr();
    printf("\n Введіть значення аргумента: ");
    scanf("%f",&x);

    // Табулювання числової послідовності
    // та обчислення суми та добутку її членів:
    printf("\n Результати табулювання числової послідовності:\n");
    clrscr();

    // Обчислення чисельника дробу
```



```

printf("\n  Результати обчислення чисельника дробу:");
k=1;
sum1=0;
do {
    n=1;
    d1=1;
    do {
        d1=d1*sin(n*x);
        n++;
    } while (n <=k);
    printf("\n  k=%3.0f      d1=%+8.5f", k,d1);
    sum1=sum1+d1;
    k++;
} while (k<11);
printf("\n");

// Обчислення знаменника дробу
printf("\n  Результати обчислення знаменника дробу:");
k=1;
sum2=0;
do {
    n=1;
    d2=1;
    do {
        d2=d2*cos(n*x);
        n++;
    } while (n <=k);
    printf("\n  k=%3.0f      d1=%+8.5f", k,d1);
    sum2=sum2+d2;
    k++;
} while (k<=11);
printf("\n");

// Обчислення та виведення на екран величини дробу:
a=sum1/sum2;
printf("\n  Значення аргумента дорівнює
x=%+8.5f;",sum1);
printf("\n  Значення чисельника дорівнює sum1=%+8.5f;",sum1);
printf("\n  Значення знаменника дорівнює sum2=%+8.5f;",sum2);
printf("\n  Значення дробу дорівнює          a=%+8.5f.",a);

getch();
}

```

Результати тестування програми:

```

Результати обчислення чисельника дробу:
k=1      d1=+0.56464
k=2      d1=+0.56464
k=3      d1=+0.56464
k=4      d1=+0.56464
k=5      d1=+0.56464
k=6      d1=-0.56464

```

```

k=7      d1=+0.56464
k=8      d1=-0.56464
k=9      d1=+0.56464
k=10     d1=-0.56464

```

Результати обчислення знаменника дробу:

```

k=1      d1=+0.56464
k=2      d1=+0.56464
k=3      d1=+0.56464
k=4      d1=+0.56464
k=5      d1=+0.56464
k=6      d1=-0.56464
k=7      d1=+0.56464
k=8      d1=-0.56464
k=9      d1=+0.56464
k=10     d1=-0.56464

```

Значення аргумента дорівнює $x=1.98735$

Значення чисельника дорівнює $sum1=+1.98735$

Значення чисельника дорівнює $sum2=1.074241$

Значення дробу дорівнює $a=+1.85001$

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Скласти, відлагодити та протестувати циклічну програму розв'язування задачі обчислення заданого виразу за допомогою оператора `do...while`, якщо параметри a (дійсного типу) та n (цілого типу) задані.

1. $\frac{1}{a} + \frac{1}{a(a+1)} + \dots + \frac{1}{a(a+1)\dots(a+n)}, \quad a \neq 0.$

2. $\sqrt{a + \sqrt{a + \dots + \sqrt{a}}}, \quad a > 1.$

3. $\sin a + \sin 2a + \dots + \sin na.$

4.
$$\frac{a}{1 + \frac{a}{3 + \frac{a}{5 + \frac{a}{\ddots \frac{a}{(n-1) + \frac{a}{n}}}}}}$$

5. $1 * 2 + 2 * 3 * 4 + \dots + n(n+1) \dots 2n.$

6. $\left(1 + \frac{1}{1^2}\right) * \left(1 + \frac{1}{2^2}\right) \dots \left(1 + \frac{1}{n^2}\right).$

7. $\sin a + \sin \sin a + \dots + \sin \sin \dots \sin a.$

8. $\sqrt{3a + \sqrt{6a + \dots + \sqrt{3(n-1)a + \sqrt{3na}}}}, \quad a > 0.$

ЛАБОРАТОРНА РОБОТА №11: “ПРОГРАМУВАННЯ ІТЕРАЦІЙНИХ ПРОЦЕСІВ МОВОЮ C/C++: ЧИСЕЛЬНЕ РОЗВ’ЯЗУВАННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ ПОЛОВИННОГО ДІЛЕННЯ.”

МЕТА РОБОТИ

1. Ознайомитися з поняттями ітераційних процесів та алгоритмів.
2. Ознайомитися з етапами чисельного розв’язування нелінійних рівнянь.
3. Ознайомитися з уточненням коренів методом половинного ділення.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_11_01` для розв’язування наступної задачі. Дано: нелінійне рівняння $f(x) = x^3 - 2.02x^2 - 5.01x + 8 = 0$. Необхідно обчислити з заданою точністю $\epsilon = 0.001$ та вивести на друк значення всі значення дійсних коренів цього рівняння, використовуючи метод половинного ділення. Циклічні процеси реалізувати відповідними операторами циклу мови C. Для обчислення значення функції $f(x)$ в довільній точці створити і використати у програмі функцію користувача.

```
/* **** */
/* lab_work_11_01. */
/* Ітераційні обчислювальні процеси. */
/* Обчислення з заданою точністю дійсних коренів */
/* нелінійних рівнянь засобами мови C. */
/* Дано: нелінійне рівняння */
/*  $f(x) = x^3 - 2.02x^2 - 5.01x + 8 = 0$ . */
/* Необхідно: */
/* обчислити з заданою точністю  $\epsilon = 0.001$  */
/* всі дійсні корені цього рівняння, */
/* використовуючи метод половинного ділення. */
/* Створити програму мовою C, */
/* використовуючи відповідні оператори циклу. */
/* Створити функцію користувача мовою C */
/* для обчислення значення даної функції в довільній точці. */
/* **** */

#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <windows.h>

float a; // Ліва межа інтервалу ізоляції всіх коренів
float b; // Права межа інтервалу ізоляції всіх коренів
float x; // Аргумент функції
float y; // Значення функції
float h; // Крок зміни аргумента
float yp; // Початкове значення функції
float ys; // Значення функції посередині інтервалу [a;b]
int k; // Номер наближення кореня
float eps=0.001; // Точність обчислення коренів

float f(float x); // Прототип функції користувача
```

```

void main() {
    clrscr();
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    printf("\nВвести ліву межу інтервалу ізоляції всіх коренів a: ");
    scanf("%f",&a);
    printf("Ввести праву межу інтервалу ізоляції всіх коренів b: ");
    scanf("%f",&b);
    printf("Ввести крок зміни аргумента h: ");
    scanf("%f",&h);

    // Етап відокремлення коренів рівняння:
    x=a;
    yp=f(a);
    k=0;
    while (x<b+h/2){
        y=f(x);
        printf("\n  x=%6.2f  y=%+6.2f",x,y);
        if (y*yp<0) {
            printf("\n\n  Маємо %d-й інтервал ізоляції коренів рівняння:
",k+1);
            printf("\n  a[%d]=%+6.2f  b[%d]=%+6.2f\n",k+1,x-h,k+1,x);
            k++;
        }
        yp=y;
        x+=h;
    }
    if (k==0)
        printf("\n\n  Рівняння не має коренів на заданому інтервалі
[%+6.2f,%+6.2f]: ",a,b);
    else
        printf("\n\n  Рівняння має %d корені на заданому інтервалі
[%+6.2f,%+6.2f]: ",k,a,b);
    printf("\n\n  Натисніть Enter, щоб перейти до уточнення коренів.");
    getch();

    // Етап уточнення коренів рівняння до заданої точності:
    printf("\n\nВвести ліву межу інтервалу ізоляції одного кореня a:
");
    scanf("%f",&a);
    printf("Ввести праву межу інтервалу ізоляції одного кореня b: ");
    scanf("%f",&b);

    k=0;
    x=(a+b)/2;
    printf("\n  %d-наближення кореня x[%d]=%+6.4f",k,k,x);
    while (fabs(b-a)>=eps){
        ys=f(a);
        y=f(x);
        if (ys*y<=0)
            b=x;
        else

```

```

        a=x;
        x=(a+b)/2;
        k++;
        printf("\n %d-наближення кореня x[%d]=%+6.4f",k,k,x);
    }

// Друк коренів рівняння:
    y=f(x);
    printf("\n\nОстаточнo: Корiнь x=%+6.4f. Значення функції
y(%+6.4f)=%+6.4f\n",x,x,y);
    getch();
}

//-----
// Ця функція користувача обчислює значення функції f(x)
float f(float x){
    return x*x*x-2.02*x*x-5.01*x+8;
}

```

Результати тестування програми:

Ввести ліву межу інтервалу ізоляції всіх коренів a: -10
Ввести праву межу інтервалу ізоляції всіх коренів b: 10
Ввести крок зміни аргумента h: 1

```

x=-10.00  y=-1143.90
x= -9.00  y=-839.53
x= -8.00  y=-593.20
x= -7.00  y=-398.91
x= -6.00  y=-250.66
x= -5.00  y=-142.45
x= -4.00  y=-68.28
x= -3.00  y=-22.15
x= -2.00  y= +1.94

```

Маємо 1-й інтервал ізоляції коренів рівняння:
a[1]= -3.00 b[1]= -2.00

```

x= -1.00  y= +9.99
x=  0.00  y= +8.00
x=  1.00  y= +1.97
x=  2.00  y= -2.10

```

Маємо 2-й інтервал ізоляції коренів рівняння:
a[2]= +1.00 b[2]= +2.00

```

x=  3.00  y= +1.79

```

Маємо 3-й інтервал ізоляції коренів рівняння:
a[3]= +2.00 b[3]= +3.00

```

x=  4.00  y=+19.64
x=  5.00  y=+57.45
x=  6.00  y=+121.22

```

$x = 7.00 \quad y = +216.95$
 $x = 8.00 \quad y = +350.64$
 $x = 9.00 \quad y = +528.29$
 $x = 10.00 \quad y = +755.90$

Рівняння має 3 корені на заданому інтервалі $[-10.00, +10.00]$:
Натисніть Enter, щоб перейти до уточнення коренів.

Ввести ліву межу інтервалу ізоляції одного кореня а: -3
Ввести праву межу інтервалу ізоляції одного кореня b: -2

0-наближення кореня $x[0] = -2.5000$
1-наближення кореня $x[1] = -2.2500$
2-наближення кореня $x[2] = -2.1250$
3-наближення кореня $x[3] = -2.0625$
4-наближення кореня $x[4] = -2.0938$
5-наближення кореня $x[5] = -2.1094$
6-наближення кореня $x[6] = -2.1172$
7-наближення кореня $x[7] = -2.1211$
8-наближення кореня $x[8] = -2.1191$
9-наближення кореня $x[9] = -2.1201$
10-наближення кореня $x[10] = -2.1206$

Остаточо: Корінь $x = -2.1206$. Значення функції $y(-2.1206) = +0.0041$

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Відокремити корені заданого за варіантом трансцендентного рівняння графічно. Перевірити, чи на кожному інтервалі ізоляції коренів дійсно знаходиться лише один корінь, та визначити його з точністю $\epsilon = 0.001$ методом половинного ділення.

№	Трансцендентне рівняння $f(x) = 0$:
1	$x - \sin x - 0.25 = 0.$
2	$\operatorname{tg}(0.58x + 0.1) = x^2.$
3	$\sqrt{x} - \cos(0.387x) = 0.5$
4	$\operatorname{tg}(0.4x + 4) = x^2.$
5	$\lg x - \frac{7}{2x + 6} = 0.$
6	$\operatorname{tg}(0.5x + 0.2) = x^2.$
7	$3x - \cos x - 1 = 0.$
8	$x + \lg x = 0.5.$
9	$\operatorname{tg}(0.5x + 0.1) = x^2.$
10	$x^2 + 4\sin x = 1.2.$
11	$\operatorname{ctg} 1.05x - x^2 = 0.$
12	$\operatorname{tg}(0.4x + 0.3) = x^2.$
13	$x \lg x - 1.2 = 2.5.$
14	$1.8x^2 - \sin 10x = 1.$
15	$\operatorname{ctg} x - \frac{x}{4} = 0.$
16	$\operatorname{tg}(0.3x + 0.4) = x^2.$
17	$x^2 - 20\sin x = 1.$

18	$ctgx - \frac{x}{3} = 0.$
19	$tg(0.47x + 0.2) = x^2.$
20	$x^2 + 4\sin x = 1.5.$
21	$ctgx - \frac{x}{2} = 0.$
22	$2x - lgx = 7.$
23	$tg(0.44x + 0.3) = x^2.$
24	$3x - \cos x - 1.3 = 0.$
25	$ctgx - \frac{x}{10} = 0.$
26	$x^2 + 4\sin x = 1.7.$
27	$tg(0.36x + 0.4) = x^2.$
28	$x + lgx = 2.5.$
29	$ctgx - \frac{x}{5} = 0.$
30	$2lgx - \frac{\pi}{2} + 1 = 0$

Завдання 2. Відокремити корені заданого за варіантом трансцендентного рівняння графічно. Перевірити, чи на кожному інтервалі ізоляції коренів дійсно знаходиться лише один корінь, та визначити його з точністю $\epsilon=0.001$ методом ітерацій.

№	Трансцендентне рівняння: $f(x) = 0$:
1	$\ln x + (x + 1)^3 = 0.$
2	$x2^x = 1.$
3	$\sqrt{x + 1} = \frac{1}{x}.$
4	$x - \cos x = 2.$
5	$3x + \cos x + 1 = 0.$
6	$x + lgx = 0.5.$
7	$2 - x = \ln x.$
8	$(x - 1)^2 = \frac{e^x}{2}.$
9	$(2 - x)e^x = 1.5.$
10	$2.2x - 2^x = 1.$
11	$x^2 + 4\sin x = 1.$
12	$2x - lgx = 7.$
13	$5x - 8\ln x = 8.$
14	$3x - e^x = 0.$
15	$x(x + 1)^2 = 1.$
16	$x = (x + 1)^3.$
17	$x^2 - 1 = \sin x$
18	$x^3 - 1 = \sin x$
19	$x = \sqrt{\lg(x + 2)}.$
20	$x^2 = \ln(x + 1).$
21	$2x + lgx = -0.5.$
22	$2x + \cos x = 0.5.$
23	$\sin 0.5x + 1 = x^2.$

24	$0.5x + \lg(x - 1) = 0.5.$
25	$\sin(0.5 + x) = 2x - 0.5.$
26	$\lg(2 + x) + 2x = 3.$
27	$\lg(1 + 2x) = 2 - x.$
28	$2\sin(x - 0.6) = 1.5 - x.$
29	$x + \lg(1 + x) = 1.5.$
30	$x + \cos x = 1.5.$

ЛАБОРАТОРНА РОБОТА №12: “ПРОГРАМУВАННЯ ІТЕРАЦІЙНИХ ПРОЦЕСІВ МОВОЮ C/C++: ЧИСЕЛЬНЕ РОЗВ’ЯЗУВАННЯ НЕЛІНІЙНИХ РІВНЯНЬ МЕТОДАМИ ХОРД ТА ДОТИЧНИХ.”

МЕТА РОБОТИ

1. Ознайомитися з поняттями ітераційних процесів та алгоритмів.
2. Ознайомитися з етапами чисельного розв’язування нелінійних рівнянь.
3. Ознайомитися з уточненням коренів методами хорд та дотичних.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_12_01` для розв’язування наступної задачі. Дано: нелінійне рівняння $f(x) = x^3 - 2.02x^2 - 5.01x + 8 = 0$. Необхідно обчислити з заданою точністю $\epsilon = 0.001$ та вивести на друк значення всі значення дійсних коренів цього рівняння, використовуючи метод половинного ділення. Циклічні процеси реалізувати відповідними операторами циклу мови C. Для обчислення значення функції $f(x)$ в довільній точці створити і використати у програмі функцію користувача.

```
/* **** */
/* lab_work_12_01. */
/* Ітераційні обчислювальні процеси. */
/* Обчислення з заданою точністю дійсних коренів */
/* нелінійних рівнянь засобами мови C. */
/* Дано: нелінійне рівняння */
/*  $f(x) = x^3 - 2.02x^2 - 5.01x + 8 = 0$ . */
/* Необхідно: */
/* обчислити з заданою точністю  $\epsilon = 0.001$  */
/* всі дійсні корені цього рівняння, */
/* використовуючи метод хорд. */
/* Створити програму мовою C, */
/* використовуючи відповідні оператори циклу. */
/* Створити функцію користувача мовою C */
/* для обчислення значення даної функції в довільній точці. */
/* **** */
```

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <windows.h>
```

```
float a; // Ліва межа інтервалу ізоляції всіх коренів
float b; // Права межа інтервалу ізоляції всіх коренів
float x; // Аргумент функції
float y; // Значення функції
float h; // Крок зміни аргумента
float xp; // Попереднє значення аргумента
float yp; // Початкове значення функції
float ys; // Значення функції посередині інтервалу [a;b]
int k; // Номер наближення кореня
float eps=0.001; // Точність обчислення коренів
```

```
float f(float x); // Прототип функції користувача
```

```

void main() {
    clrscr();
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    printf("\nВвести ліву межу інтервалу ізоляції всіх коренів a: ");
    scanf("%f",&a);
    printf("Ввести праву межу інтервалу ізоляції всіх коренів b: ");
    scanf("%f",&b);
    printf("Ввести крок зміни аргумента h: ");
    scanf("%f",&h);

    // Етап відокремлення коренів рівняння:
    x=a;
    yp=f(a);
    k=0;
    while (x<b+h/2) {
        y=f(x);
        printf("\n  x=%6.2f  y=%+6.2f",x,y);
        if (y*yp<0) {
            printf("\n\n  Маємо %d-й інтервал ізоляції коренів рівняння:
",k+1);
            printf("\n  a[%d]=%+6.2f  b[%d]=%+6.2f\n",k+1,x-h,k+1,x);
            k++;
        }
        yp=y;
        x+=h;
    }
    if (k==0)
        printf("\n\n  Рівняння не має коренів на заданому інтервалі
[%+6.2f,%+6.2f]: ",a,b);
    else
        printf("\n\n  Рівняння має %d корені на заданому інтервалі
[%+6.2f,%+6.2f]: ",k,a,b);
    printf("\n  Натисніть Enter, щоб перейти до уточнення коренів.");
    getch();

    // Етап уточнення коренів рівняння до заданої точності:
    printf("\n\nВвести ліву межу інтервалу ізоляції одного кореня a:
");
    scanf("%f",&a);
    printf("Ввести праву межу інтервалу ізоляції одного кореня b: ");
    scanf("%f",&b);

    k=0;
    xp=b;
    x=b;
    printf("\n  %d-наближення кореня x[%d]=%+6.4f",k,k,x);
    do {
        k++;
        xp=x;
        x=xp- (f(xp) / (f(xp)-f(a))) * (xp-a);
    }

```

```

    printf("\n %d-наближення кореня x[%d]=%+6.4f", k, k, x);
} while (fabs(xp-x)>=eps);

// Друк коренів рівняння:
y=f(x);
printf("\n\nОстаточно: Корінь x=%+6.4f. Значення функції
y(%+6.4f)=%+6.4f\n", x, x, y);
getch();
}

//-----
// Ця функція користувача обчислює значення функції f(x)
float f(float x){
    return x*x*x-2.02*x*x-5.01*x+8;
}

```

Результати тестування програми:

Ввести ліву межу інтервалу ізоляції всіх коренів a: -10
Ввести праву межу інтервалу ізоляції всіх коренів b: 10
Ввести крок зміни аргумента h: 1

```

x=-10.00  y=-1143.90
x= -9.00  y=-839.53
x= -8.00  y=-593.20
x= -7.00  y=-398.91
x= -6.00  y=-250.66
x= -5.00  y=-142.45
x= -4.00  y=-68.28
x= -3.00  y=-22.15
x= -2.00  y= +1.94

```

Маємо 1-й інтервал ізоляції коренів рівняння:
a[1]= -3.00 b[1]= -2.00

```

x= -1.00  y= +9.99
x=  0.00  y= +8.00
x=  1.00  y= +1.97
x=  2.00  y= -2.10

```

Маємо 2-й інтервал ізоляції коренів рівняння:
a[2]= +1.00 b[2]= +2.00

```

x=  3.00  y= +1.79

```

Маємо 3-й інтервал ізоляції коренів рівняння:
a[3]= +2.00 b[3]= +3.00

```

x=  4.00  y=+19.64
x=  5.00  y=+57.45
x=  6.00  y=+121.22
x=  7.00  y=+216.95
x=  8.00  y=+350.64

```

```
x= 9.00 y=+528.29
x= 10.00 y=+755.90
```

Рівняння має 3 корені на заданому інтервалі [-10.00,+10.00]:
Натисніть Enter, щоб перейти до уточнення коренів.

Ввести ліву межу інтервалу ізоляції одного кореня a: -3
Ввести праву межу інтервалу ізоляції одного кореня b: -2

```
0-наближення кореня x[0]=-2.0000
1-наближення кореня x[1]=-2.0805
2-наближення кореня x[2]=-2.1077
3-наближення кореня x[3]=-2.1166
4-наближення кореня x[4]=-2.1195
5-наближення кореня x[5]=-2.1204
```

Остаточо: Корінь $x = -2.1204$. Значення функції $y(-2.1204) = +0.0076$

2. Ввести, відлагодити та протестувати програму і зберегти її під назвою *lab_work_12_02* для розв'язування наступної задачі. Дано: нелінійне рівняння $f(x) = x^3 - 2.02x^2 - 5.01x + 8 = 0$ Необхідно обчислити з заданою точністю $\epsilon = 0.001$ та вивести на друк значення всі значення дійсних коренів цього рівняння, використовуючи метод дотичних. Циклічні процеси реалізувати відповідними операторами циклу мови C. Для обчислення значення функції $f(x)$ в довільній точці створити і використати у програмі функцію користувача.

```
/* **** */
/* lab_work_12_02. */
/* Ітераційні обчислювальні процеси. */
/* Обчислення з заданою точністю дійсних коренів */
/* нелінійних рівнянь засобами мови C. */
/* Дано: нелінійне рівняння */
/*  $f(x) = x^3 - 2.02x^2 - 5.01x + 8 = 0$ . */
/* Необхідно: */
/* обчислити з заданою точністю  $\epsilon = 0.001$  */
/* всі дійсні корені цього рівняння, */
/* використовуючи метод дотичних. */
/* Створити програму мовою C, */
/* використовуючи відповідні оператори циклу. */
/* Створити функцію користувача мовою C */
/* для обчислення значення даної функції в довільній точці. */
/* **** */
```

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <windows.h>
```

```
float a; // Ліва межа інтервалу ізоляції всіх коренів
float b; // Права межа інтервалу ізоляції всіх коренів
float x; // Аргумент функції
float y; // Значення функції
float h; // Крок зміни аргумента
```

```

float xp;    // Попереднє значення аргумента
float yp;    // Початкове значення функції
float ys;    // Значення функції посередині інтервалу [a;b]
int k;       // Номер наближення кореня
float eps=0.001; // Точність обчислення коренів

float f(float x); // Прототип функції користувача
float ff(float x); // Прототип функції користувача

void main() {
    clrscr();
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    printf("\nВвести ліву межу інтервалу ізоляції всіх коренів a: ");
    scanf("%f",&a);
    printf("Ввести праву межу інтервалу ізоляції всіх коренів b: ");
    scanf("%f",&b);
    printf("Ввести крок зміни аргумента h: ");
    scanf("%f",&h);

    // Етап відокремлення коренів рівняння:
    x=a;
    yp=f(a);
    k=0;
    while (x<b+h/2) {
        y=f(x);
        printf("\n  x=%6.2f  y=%+6.2f",x,y);
        if (y*yp<0) {
            printf("\n\n  Маємо %d-й інтервал ізоляції коренів рівняння:
",k+1);
            printf("\n  a[%d]=%+6.2f  b[%d]=%+6.2f\n",k+1,x-h,k+1,x);
            k++;
        }
        yp=y;
        x+=h;
    }
    if (k==0)
        printf("\n\n  Рівняння не має коренів на заданому інтервалі
[%+6.2f,%+6.2f]: ",a,b);
    else
        printf("\n\n  Рівняння має %d корені на заданому інтервалі
[%+6.2f,%+6.2f]: ",k,a,b);
    printf("\n  Натисніть Enter, щоб перейти до уточнення коренів.");
    getch();

    // Етап уточнення коренів рівняння до заданої точності:
    printf("\n\nВвести ліву межу інтервалу ізоляції одного кореня a:
");
    scanf("%f",&a);
    printf("Ввести праву межу інтервалу ізоляції одного кореня b: ");
    scanf("%f",&b);

```

```

k=0;
xp=a;
x=a;
printf("\n %d-наближення кореня x[%d]=%+6.4f",k,k,x);
do {
    k++;
    xp=x;
    x=xp-f(xp)/ff(xp);
    printf("\n %d-наближення кореня x[%d]=%+6.4f",k,k,x);
} while (fabs(xp-x)>=eps);

// Друк коренів рівняння:
y=f(x);
printf("\n\nОстаточнo: Корінь x=%+6.4f. Значення функції
y(%+6.4f)=%+6.4f\n",x,x,y);
getch();
}

//-----
// Ця функція користувача обчислює значення функції f(x)
float f(float x){
    return x*x*x-2.02*x*x-5.01*x+8;
}

//-----
// Ця функція користувача обчислює значення похідної функції f(x)
float ff(float x){
    return 3.*x*x-4.04*x-5.01;
}

```

Результати тестування програми:

Ввести ліву межу інтервалу ізоляції всіх коренів a: -10
Ввести праву межу інтервалу ізоляції всіх коренів b: 10
Ввести крок зміни аргумента h: 1

```

x=-10.00  y=-1143.90
x= -9.00  y=-839.53
x= -8.00  y=-593.20
x= -7.00  y=-398.91
x= -6.00  y=-250.66
x= -5.00  y=-142.45
x= -4.00  y=-68.28
x= -3.00  y=-22.15
x= -2.00  y= +1.94

```

Маємо 1-й інтервал ізоляції коренів рівняння:
a[1]= -3.00 b[1]= -2.00

```

x= -1.00  y= +9.99
x=  0.00  y= +8.00
x=  1.00  y= +1.97
x=  2.00  y= -2.10

```

Маємо 2-й інтервал ізоляції коренів рівняння:
 $a[2] = +1.00$ $b[2] = +2.00$

$x = 3.00$ $y = +1.79$

Маємо 3-й інтервал ізоляції коренів рівняння:
 $a[3] = +2.00$ $b[3] = +3.00$

$x = 4.00$ $y = +19.64$
 $x = 5.00$ $y = +57.45$
 $x = 6.00$ $y = +121.22$
 $x = 7.00$ $y = +216.95$
 $x = 8.00$ $y = +350.64$
 $x = 9.00$ $y = +528.29$
 $x = 10.00$ $y = +755.90$

Рівняння має 3 корені на заданому інтервалі $[-10.00, +10.00]$:
 Натисніть Enter, щоб перейти до уточнення коренів.

Ввести ліву межу інтервалу ізоляції одного кореня a : -3
 Ввести праву межу інтервалу ізоляції одного кореня b : -2

0-наближення кореня $x[0] = -3.0000$
 1-наближення кореня $x[1] = -2.3506$
 2-наближення кореня $x[2] = -2.1430$
 3-наближення кореня $x[3] = -2.1211$
 4-наближення кореня $x[4] = -2.1208$

Остаточо: Корінь $x = -2.1208$. Значення функції $y(-2.1208) = -0.0000$

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Відокремити корені заданого за варіантом алгебраїчного рівняння аналітично. Перевірити, чи на кожному інтервалі ізоляції коренів дійсно знаходиться лише один корінь, та визначити його з точністю $\epsilon = 0.001$ методом хорд. Програму обчислень зберегти в m-файлі під назвою lab_work_09_02.

№	Алгебраїчне рівняння: $f(x) = 0$
1	$2x^3 - 3x^2 + 12x - 5 = 0.$
2	$x^3 - 3x^2 + 3 = 0.$
3	$x^3 + 3x^2 - 24x - 10 = 0.$
4	$2x^3 + 9x^2 - 21 = 0.$
5	$x^3 + 3x^2 - 2 = 0.$
6	$x^3 + 3x^2 - 24x + 10 = 0.$
7	$2x^3 + 9x^2 - 10 = 0.$
8	$x^3 + 3x^2 - 3 = 0.$
9	$x^3 - 3x^2 - 24x - 5 = 0.$
10	$x^3 - 12x - 5 = 0.$
11	$2x^3 - 3x^2 - 12x + 12 = 0.$
12	$x^3 - 3x^2 + 1.5 = 0.$

13	$x^3 + 3x^2 - 24x - 3 = 0.$
14	$x^3 - 3x^2 - 24x + 10 = 0.$
15	$2x^3 + 9x^2 - 4 = 0.$
16	$x^3 + 3x^2 - 1 = 0.$
17	$x^3 - 3x^2 - 24x - 3 = 0.$
18	$x^3 - 12x + 6 = 0.$
19	$2x^3 - 3x^2 - 12x + 10 = 0.$
20	$x^3 - 3x^2 + 2.5 = 0.$
21	$x^3 + 3x^2 - 3.5 = 0.$
22	$x^3 - 3x^2 - 24x - 8 = 0.$
23	$x^3 - 12x + 10 = 0.$
24	$2x^3 - 3x^2 - 12x + 1 = 0.$
25	$x^3 - 4x^2 + 2 = 0.$
26	$x^3 + 3x^2 - 24x + 1 = 0.$
27	$2x^3 + 9x^2 - 6 = 0.$
28	$x^3 - 12x - 10 = 0.$
29	$2x^3 - 3x^2 - 12x + 8 = 0.$
30	$x^3 - 3x^2 + 3.5 = 0.$

Завдання 2. Відокремити корені заданого за варіантом алгебраїчного рівняння аналітично. Перевірити, чи на кожному інтервалі ізоляції коренів дійсно знаходиться лише один корінь, та визначити його з точністю $\epsilon=0.001$ методом хорд. Програму обчислень зберегти в m-файлі під назвою lab_work_09_03.

№	Алгебраїчне рівняння: $f(x) = 0$:
1	$x^3 - 3x^2 + 9x - 8 = 0.$
2	$x^3 - 6x - 8 = 0.$
3	$x^3 - 3x^2 + 6x + 3 = 0.$
4	$x^3 - 0.1x^2 + 0.4x - 1.5 = 0.$
5	$x^3 - 3x^2 + 9x + 2 = 0.$
6	$x^3 + x - 5 = 0.$
7	$x^3 + 0.2x^2 + 0.5x - 1.2 = 0.$
8	$x^3 + 3x + 1 = 0.$
9	$x^3 + 0.2x^2 + 0.5x - 2 = 0.$
10	$x^3 - 3x^2 + 12x - 9 = 0.$
11	$x^3 - 0.2x^2 + 0.3x - 1.2 = 0.$
12	$x^3 - 3x^2 + 6x - 2 = 0.$
13	$x^3 - 0.1x^2 + 0.4x - 1.5 = 0.$
14	$x^3 + 3x^2 + 6x - 1 = 0.$
15	$x^3 + 0.1x^2 + 0.4x - 1.2 = 0.$
16	$x^3 + 4x - 6 = 0.$
17	$x^3 + 0.2x^2 + 0.5x + 0.8 = 0.$
18	$x^3 - 3x^2 + 12x - 12 = 0.$
19	$x^3 - 0.2x^2 + 0.3x + 1.2 = 0.$
20	$x^3 - 2x + 4 = 0.$
21	$x^3 - 0.2x^2 + 0.5x - 1.4 = 0.$
22	$x^3 - 3x^2 + 6x - 5 = 0.$
23	$x^3 - 0.1x^2 + 0.4x + 1.2 = 0.$
24	$x^3 - 0.2x^2 + 0.5x - 1 = 0.$

25	$x^3 + 3x^2 + 12x + 3 = 0.$
26	$x^3 - 0.1x^2 + 0.4x + 2 = 0.$
27	$x^3 - 0.2x^2 + 0.4x - 1.4.$
28	$x^3 + 0.4x^2 + 0.6x - 1.6 = 0.$
29	$x^3 + x - 3 = 0.$
30	$x^3 - 0.2x^2 + 0.5x + 1.4 = 0.$

ЛАБОРАТОРНА РОБОТА №13: “ОБЧИСЛЕННЯ З ЗАДАНОЮ ТОЧНІСТЮ СУМИ ЧЛЕНІВ ЗНАКОЗМІННОГО РЯДУ.”

МЕТА РОБОТИ

1. Ознайомитися з методикою побудови рекурентних формул.
2. Ознайомитися з методикою обчислення з заданою точністю суми знакозмінного ряду.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_13_01` для розв’язування наступної задачі. Дано: знакозмінний ряд, загальний член якого дорівнює $a[n] = ((-1)^{(n+1)}) * (x^{(2*n-1)}) / (2n-1)!$, $n=1, 2, \dots$ і відповідна цьому ряду функція $y = \sin(x)$, де x – довільне дійсне число (аргумент ряду і функції). Необхідно: обчислити з заданою точністю $\text{eps}=0.001$ суму sum членів ряду і відповідне значення функції $y = \sin(x)$ у заданій точці $x=0.2$; побудувати рекурентну формулу для обчислення значення поточного члена ряду. Циклічні процеси реалізувати відповідними оператором циклу `<while>` мови C.

```
/* **** */
/* lab_work_13_01. */
/* Ітераційні обчислювальні процеси. */
/* Обчислення з заданою точністю суми членів */
/* знакозмінного ряду засобами мови C. */
/* Дано: знакозмінний ряд, загальний член якого дорівнює */
/*  $a[n] = ((-1)^{(n+1)}) * (x^{(2*n-1)}) / (2n-1)!$ ,  $n=1, 2, \dots$  */
/* і відповідна цьому ряду функція  $y = \sin(x)$ , */
/* де  $x$  – довільне дійсне число */
/* (аргумент ряду і функції.) */
/* Необхідно: */
/* обчислити з заданою точністю суму  $\text{sum}$  членів ряду */
/* і відповідне значення функції  $y = \sin(x)$  */
/* в заданій точці  $x=0.2$ ; */
/* побудувати рекурентну формулу для обчислення значення */
/* поточного члена ряду. */
/* Створити програму засобами мови C, */
/* використовуючи оператор <while>. */
/* **** */
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <windows.h>

float x;          // Аргумент функції та ряду
float y;          // Значення функції
float s=0;        // Сума  $\text{sum}$  членів ряду
float a;          // Значення поточного члена ряду
int n=1;          // Номер поточного члена ряду
float eps=0.001 ; // Точність обчислення суми ряду

void main() {
    clrscr();
```

```

SetConsoleCP(1251);
SetConsoleOutputCP(1251);

// Введення аргумента:
printf("\n Введіть значення аргумента x:  ");
scanf("%f",&x);

a=x; // Нульове наближення поточного члена ряду
s=x; // Нульове наближення суми членів ряду

printf("\n  Значення поточного члена ряду:",n,a);
printf("\n  a(%d)=%+6.4f\n",n,a);

while (fabs(a)>=eps/2)    {
    a=-(x/(2*n))*(x/(2*n+1))*a; // Рекурентна формула
    printf("  a(%d)=%+6.4f\n",n+1,a);
    s=s+a;
    n+=1;
}

y=sin(x);

printf("\n Сума членів знакозмінного ряду дорівнює s=%8.4f",s);
printf("\n Значення функції y=sinx дорівнює y=%8.4f",y);
printf("\n Використано в обчисленнях таку кількість членів ряду:
n=%d\n",n);
getch();
}

```

Результати тестування програми:

Введіть значення аргумента x: 0.2

Значення поточного члена ряду:

a(1)=+0.2000

a(2)=-0.0013

a(3)=+0.0000

Сума членів знакозмінного ряду дорівнює s= 0.1987

Значення функції y=sinx дорівнює y= 0.1987

Використано в обчисленнях таку кількість членів ряду: n=3

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Обчислити у заданій точці x_0 з точністю $\epsilon=0.001$ значення заданої за

варіантом функції $y = f(x)$, скориставшись формулою розкладу цієї функції у відповідний знакозмінний ряд. Перед складанням програми побудувати рекурентні формули для обчислення значення члена поточного члена ряду.

№	Функція	Знакозмінний ряд, що відповідає заданій функції	Область	x_0
1	$y = \sin x$	$\frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.1
2	$y = \sin x^2$	$\frac{x^2}{1!} - \frac{x^6}{3!} + \frac{x^{10}}{5!} - \dots + (-1)^{n+1} \frac{x^{4n-2}}{(2n-1)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.2
3	$y = \sin x^3$	$\frac{x^3}{1!} - \frac{x^9}{3!} + \frac{x^{15}}{5!} - \dots + (-1)^{n+1} \frac{x^{6n-3}}{(2n-1)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.3
4	$y = \frac{\sin x}{x}$	$\frac{1}{1!} - \frac{x^2}{3!} + \frac{x^4}{5!} - \dots + (-1)^{n+1} \frac{x^{2n-2}}{(2n-1)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$ $x \neq 0$	0.4
5	$y = \frac{\sin x}{x^2}$	$\frac{1}{1!x} - \frac{x}{3!} + \frac{x^3}{5!} - \dots + (-1)^{n+1} \frac{x^{2n-3}}{(2n-1)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$ $x \neq 0$	0.5
6	$y = e^{-x}$	$\frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.1
7	$y = e^{-x^2}$	$1 - \frac{x^2}{1!} + \frac{x^4}{2!} - \dots + (-1)^n \frac{x^{2n}}{n!} + \dots; \quad n = 0, 1, \dots$	$ x < \infty$	0.2
8	$y = e^{-x^3}$	$1 - \frac{x^3}{1!} + \frac{x^6}{2!} - \dots + (-1)^n \frac{x^{3n}}{n!} + \dots; \quad n = 0, 1, \dots$	$ x < \infty$	0.3
9	$y = xe^{-x}$	$x - \frac{x^2}{1!} + \frac{x^3}{2!} - \dots + (-1)^n \frac{x^{n+1}}{n!} + \dots; \quad n = 0, 1, \dots$	$ x < \infty$	0.4
10	$y = x^2 e^{-x}$	$x^2 - \frac{x^3}{1!} + \frac{x^4}{2!} - \dots + (-1)^n \frac{x^{n+2}}{n!} + \dots; \quad n = 0, 1, \dots$	$ x < \infty$	0.5
11	$y = \ln(x+1)$	$x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n+1} \frac{x^n}{n} + \dots; \quad n = 1, 2, \dots$	$x > -1$	0.1
12	$y = \ln(x^2+1)$	$x^2 - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n+1} \frac{x^n}{n} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.2
13	$y = \ln(x^3+1)$	$x^3 - \frac{x^6}{2} + \frac{x^9}{3} - \dots + (-1)^{n+1} \frac{x^{3n}}{n} + \dots; \quad n = 1, 2, \dots$	$x > -1$	0.3
14	$y = \frac{\ln(x+1)}{x}$	$x - \frac{x^4}{2} + \frac{x^6}{3} - \dots + (-1)^{n+1} \frac{x^n}{n} + \dots; \quad n = 1, 2, \dots, \dots$	$x > -1$; $x \neq 0$	0.4
15	$y = \frac{\ln(x^2+1)}{x}$	$1 - \frac{x}{2} + \frac{x^2}{3} - \dots + (-1)^{n+1} \frac{x^{n-1}}{n} + \dots; \quad n = 1, 2, \dots, \dots$	$ x < \infty$; $x \neq 0$	0.5
16	$y = \frac{1}{1+x}$	$1 - x + x^2 - \dots + (-1)^n x^n + \dots; \quad n = 1, 2, \dots$	$x \neq -1$	0.1
17	$y = \frac{1}{1+x^2}$	$1 - x^2 + x^4 - \dots + (-1)^n x^{2n} + \dots; \quad n = 0, 1, \dots$	$ x < \infty$	0.2
18	$y = \frac{1}{1+x^3}$	$1 - x^3 + x^6 - \dots + (-1)^n x^{3n} + \dots; \quad n = 0, 1, \dots$	$x \neq -1$	0.3
19	$y = \frac{x}{1+x}$	$x - x^2 + x^3 - \dots + (-1)^n x^{n+1} + \dots; \quad n = 0, 1, \dots$	$x \neq -1$	0.4
20	$y = \frac{x^2}{1+x}$	$x^2 - x^3 + x^4 - \dots + (-1)^n x^{n+2} + \dots; \quad n = 0, 1, \dots$	$x \neq -1$	0.5
21	$y = \arctg x$	$x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)} + \dots; \quad n = 1, 2, \dots$	$ x < 1$	0.1
22	$y = \arctg x^2$	$x^2 - \frac{x^6}{3} + \frac{x^{10}}{5} - \dots + (-1)^{n+1} \frac{x^{4n-2}}{(2n-1)} + \dots; \quad n = 1, 2, \dots$	$ x < 1$	0.2

№	Функція	Знакозмінний ряд, що відповідає заданій функції	Область	x_0
23	$y = \arctg x^3$	$x^3 - \frac{x^9}{3} + \frac{x^{15}}{5} - \dots + (-1)^{n+1} \frac{x^{6n-3}}{(2n-1)} + \dots; \quad n = 1, 2, \dots$	$ x < 1$	0.3
24	$y = x \arctg x$	$x^2 - \frac{x^4}{3} + \frac{x^6}{5} - \dots + (-1)^{n+1} \frac{x^{2n}}{(2n-1)} + \dots; \quad n = 1, 2, \dots$	$ x < 1$	0.4
25	$y = \frac{\arctg x}{x}$	$1 - \frac{x^2}{3} + \frac{x^4}{5} - \dots + (-1)^{n+1} \frac{x^{2n-2}}{(2n-1)} + \dots; \quad n = 1, 2, \dots$	$ x < 1;$ $x \neq 0.$	0.5
26	$y = \cos x$	$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^{n+1} \frac{x^{2n-2}}{(2n-2)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.1
27	$y = \cos x^2$	$1 - \frac{x^4}{2!} + \frac{x^8}{4!} - \dots + (-1)^{n+1} \frac{x^{4n-4}}{(2n-2)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.2
28	$y = \cos x^3$	$1 - \frac{x^6}{2!} + \frac{x^{12}}{4!} - \dots + (-1)^{n+1} \frac{x^{6n-6}}{(2n-2)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.3
29	$y = x \cos x$	$x - \frac{x^3}{2!} + \frac{x^5}{4!} - \dots + (-1)^{n+1} \frac{x^{2n-1}}{(2n-2)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.4
30	$y = x^2 \cos x$	$x^2 - \frac{x^4}{2!} + \frac{x^6}{4!} - \dots + (-1)^{n+1} \frac{x^{2n}}{(2n-2)!} + \dots; \quad n = 1, 2, \dots$	$ x < \infty$	0.5

ЛАБОРАТОРНА РОБОТА №14: “ОБЧИСЛЕННЯ З ЗАДАНОЮ ТОЧНІСТЮ СУМИ ЧЛЕНІВ ЗНАКОПОСТІЙНОГО РЯДУ”

МЕТА РОБОТИ

1. Навчитися будувати рекурентні співвідношення для обчислення поточного члену та суми знакопостійного ряду.
2. Ознайомитися з технологією створення ітераційних програм засобами мови C/C++ для обчислення суми членів знакопостійного ряду.

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму і зберегти її під назвою `lab_work_14_01` для розв’язування наступної задачі. Дано: знакопостійний ряд, загальний член якого дорівнює $a[n] = (x^n)/n!$, $n=1,2,\dots$; залишковий член ряду дорівнює $r[n] = (x^n)/(n+1)!$, $n=1,2,\dots$; відповідна цьому ряду функція $y = \exp(x)$, де x – довільне дійсне число (аргумент ряду і функції). Необхідно: обчислити з заданою точністю $\text{eps}=0.001$ суму sum членів ряду і відповідне значення функції $y = \exp(x)$ у заданій точці $x=0.5$; побудувати рекурентну формули для обчислення значення поточного та залишкового членів ряду. Циклічні процеси реалізувати відповідними оператором циклу `<while>` мови C.

```
/* **** */
/* lab_work_14_01. */
/* Ітераційні обчислювальні процеси. */
/* Обчислення з заданою точністю суми членів */
/* знакопостійного ряду засобами мови C. */
/* Дано: знакопостійний ряд, загальний член якого дорівнює */
/*  $a[n] = (x^n)/n!$ ,  $n=1,2,\dots$ ; */
/* залишковий член ряду дорівнює */
/*  $r[n] = (x^n)/(n+1)!$ ,  $n=1,2,\dots$ ; */
/* відповідна цьому ряду функція  $y = \exp(x)$ , */
/* де  $x$  – довільне дійсне число */
/* (аргумент ряду і функції.) */
/* Необхідно: */
/* обчислити з заданою точністю суму  $\text{sum}$  членів ряду */
/* і відповідне значення функції  $y = \exp(x)$  */
/* в заданій точці  $x=0.5$ ; */
/* побудувати рекурентні формули для обчислення значення */
/* поточного залишкового членів ряду. */
/* Створити програму засобами мови C, */
/* використовуючи оператор <while>. */
/* **** */

#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <windows.h>

float y;          // Значення функції
float s=0;        // Сума членів  $\text{sum}$  знакопостійного ряду
float a;          // Значення поточного члена ряду
int n=0;          // Номер поточного члена ряду
```

```

float eps=0.001 ; // Точність обчислення суми sum членів ряду
float x;          // Аргумент функції тв ряду
float r;          // Залишковий член ряду

void main() {
    clrscr();
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    printf("\n Введіть задане значення аргумента x: ");
    scanf("%f",&x);

    a=1; // Нульове наближення поточного члена ряду
    r=1; // Нульове наближення залишкового члена ряду
    s=1; // Початкове значення суми ряду

    printf("\n Значення поточного та залишкового членів ряду:");
    printf("\n a(%d)=+8.4f r(%d)=+8.4f",n,a,n,r);

    while (fabs(r)>=eps/2) {
        a=(x/(n+1))*a;
        r=((x/(n+1))*(n+2))*r;
        printf("\n a(%d)=+8.4f r(%d)=+8.4f",n+1,a,n+1,r);
        s=s+a;
        n+=1;
    }

    y=exp(x);
    printf("\n");
    printf("\n Значення суми членів знакопостійного ряду
s=%8.4f",s);
    printf("\n Значення функції y=%8.4f",y);
    printf("\n Використано в обчисленнях таку кількість членів ряду:
n=%d\n",n);

    getch();
}

```

Результати тестування програми:

Введіть задане значення аргумента x: 0.5

Значення поточного та залишкового членів ряду:

a(0)= +1.0000	r(0)= +1.0000
a(1)= +0.5000	r(1)= +1.0000
a(2)= +0.1250	r(2)= +0.7500
a(3)= +0.0208	r(3)= +0.5000
a(4)= +0.0026	r(4)= +0.3125
a(5)= +0.0003	r(5)= +0.1875
a(6)= +0.0000	r(6)= +0.1094
a(7)= +0.0000	r(7)= +0.0625
a(8)= +0.0000	r(8)= +0.0352
a(9)= +0.0000	r(9)= +0.0195

a(10)= +0.0000	r(10)= +0.0107
a(11)= +0.0000	r(11)= +0.0059
a(12)= +0.0000	r(12)= +0.0032
a(13)= +0.0000	r(13)= +0.0017
a(14)= +0.0000	r(14)= +0.0009
a(15)= +0.0000	r(15)= +0.0005

Значення суми членів знакопостійного ряду s= 1.6487

Значення функції y= 1.6487

Використано в обчисленнях таку кількість членів ряду: n=15

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Обчислити у заданій точці $x_0 = 0.5$ з точністю epsilon=0.001 значення заданої

за варіантом функції $y = f(x)$, скориставшись формулою розкладу цієї функції у відповідний знакопостійний ряд. Перед складанням програми побудувати рекурентні формули для обчислення значень поточного та залишкового членів ряду. Програму обчислень зберегти в м-файлі під назвою lab_work_12_02.

№	Функція	Знакозмінний ряд, що відповідає заданій функції	Залишковий член ряду
1	$y = shx = \frac{e^x - e^{-x}}{2}$	$x + \frac{x^3}{3!} + \dots + \frac{x^{2n-1}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n-1}}{3(2n-1)!}$
2	$y = xshx = \frac{x(e^x - e^{-x})}{2}$	$x^2 + \frac{x^4}{3!} + \dots + \frac{x^{2n}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n}}{3(2n-1)!}$
3	$y = x^2shx = \frac{x^2(e^x - e^{-x})}{2}$	$x^3 + \frac{x^5}{3!} + \dots + \frac{x^{2n+1}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n+1}}{3(2n-1)!}$
4	$y = x^3shx = \frac{x^3(e^x - e^{-x})}{2}$	$x^4 + \frac{x^6}{3!} + \dots + \frac{x^{2n+2}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n+2}}{3(2n-1)!}$
5	$y = x^4shx = \frac{x^4(e^x - e^{-x})}{2}$	$x^5 + \frac{x^7}{3!} + \dots + \frac{x^{2n+3}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n+3}}{3(2n-1)!}$
6	$y = x^5shx = \frac{x^5(e^x - e^{-x})}{2}$	$x^6 + \frac{x^8}{3!} + \dots + \frac{x^{2n+4}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n+4}}{3(2n-1)!}$
7	$y = chx = \frac{e^x + e^{-x}}{2}$	$1 + \frac{x^2}{2!} + \dots + \frac{x^{2n}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n}}{3(2n)!}$
8	$y = xchx = \frac{x(e^x + e^{-x})}{2}$	$x + \frac{x^3}{2!} + \dots + \frac{x^{2n+1}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n+1}}{3(2n)!}$
9	$y = x^2chx = \frac{x^2(e^x + e^{-x})}{2}$	$x^2 + \frac{x^4}{2!} + \dots + \frac{x^{2n+2}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n+2}}{3(2n)!}$
10	$y = x^3chx = \frac{x^3(e^x + e^{-x})}{2}$	$x^3 + \frac{x^5}{2!} + \dots + \frac{x^{2n+3}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n+3}}{3(2n)!}$
11	$y = x^4chx = \frac{x^4(e^x + e^{-x})}{2}$	$x^4 + \frac{x^6}{2!} + \dots + \frac{x^{2n+4}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n+4}}{3(2n)!}$
12	$y = x^5chx = \frac{x^5(e^x + e^{-x})}{2}$	$x^5 + \frac{x^7}{2!} + \dots + \frac{x^{2n+5}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n+5}}{3(2n)!}$
13	$y = e^x$	$1 + \frac{x}{1!} + \dots + \frac{x^n}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^n}{(n+1)!}$

№	Функція	Знакозмінний ряд, що відповідає заданій функції	Залишковий член ряду
14	$y = xe^x$	$x + \frac{x^2}{2!} + \dots + \frac{x^{n+1}}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^{n+1}}{(n+1)!}$
15	$y = x^2 e^x$	$x^2 + \frac{x^3}{2!} + \dots + \frac{x^{n+2}}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^{n+2}}{(n+1)!}$
16	$y = x^3 e^x$	$x^3 + \frac{x^4}{2!} + \dots + \frac{x^{n+3}}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^{n+3}}{(n+1)!}$
17	$y = x^4 e^x$	$x^4 + \frac{x^5}{2!} + \dots + \frac{x^{n+4}}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^{n+4}}{(n+1)!}$
18	$y = x^5 e^x$	$x^5 + \frac{x^6}{2!} + \dots + \frac{x^{n+5}}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^{n+5}}{(n+1)!}$
19	$y = \frac{\sinh x}{x}$	$1 + \frac{x^2}{3!} + \dots + \frac{x^{2n-2}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n-2}}{3(2n-1)!}$
20	$y = \frac{\sinh x}{x^2}$	$\frac{1}{x} + \frac{x}{3!} + \dots + \frac{x^{2n-3}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n-1}}{3(2n-1)!}$
21	$y = \frac{\sinh x}{x^3}$	$\frac{1}{x^2} + \frac{1}{3!} + \dots + \frac{x^{2n-4}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n-3}}{3(2n-1)!}$
22	$y = \frac{\sinh x}{x^4}$	$\frac{1}{x^3} + \frac{1}{x3!} + \dots + \frac{x^{2n-5}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n-3}}{3(2n-1)!}$
23	$y = \frac{\sinh x}{x^5}$	$\frac{1}{x^4} + \frac{1}{x^2 3!} + \dots + \frac{x^{2n-6}}{(2n-1)!} + \dots; n = 1, 2, \dots$	$\frac{x^{2n-3}}{3(2n-1)!}$
24	$y = \frac{\cosh x}{x}$	$\frac{1}{x} + \frac{x}{2!} + \dots + \frac{x^{2n-1}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n-1}}{3(2n)!}$
25	$y = \frac{\cosh x}{x^2}$	$\frac{1}{x^2} + \frac{1}{2!} + \dots + \frac{x^{2n-2}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n-2}}{3(2n)!}$
26	$y = \frac{\cosh x}{x^3}$	$\frac{1}{x^3} + \frac{1}{x2!} + \dots + \frac{x^{2n-3}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n-3}}{3(2n)!}$
27	$y = \frac{\cosh x}{x^4}$	$\frac{1}{x^4} + \frac{1}{x^2 2!} + \dots + \frac{x^{2n-4}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n-4}}{3(2n)!}$
28	$y = \frac{\cosh x}{x^5}$	$\frac{1}{x^5} + \frac{1}{x^3 2!} + \dots + \frac{x^{2n-5}}{(2n)!} + \dots; n = 0, 1, \dots$	$\frac{2x^{2n-5}}{3(2n)!}$
29	$y = \frac{e^x}{x}$	$\frac{1}{x} + \frac{1}{1!} + \dots + \frac{x^{n-1}}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^{n-1}}{(n+1)!}$
30	$y = \frac{e^x}{x^2}$	$\frac{1}{x^2} + \frac{1}{x1!} + \dots + \frac{x^{n-2}}{n!} + \dots; n = 0, 1, \dots$	$\frac{x^{n-2}}{(n+1)!}$

ЛАБОРАТОРНА РОБОТА №15: “РЕКУРСИВНІ ФУНКЦІЇ МОВОЮ C/C++”

МЕТА РОБОТИ

1. Ознайомитися з поняттям рекурсії у мові C/C++.
2. Ознайомитися з технологією створення рекурсивних функцій засобами мови C/C++

ХІД РОБОТИ

1. Ввести, відлагодити та протестувати програму зберегти її під назвою `lab_work_15_01` для розв’язування такої задачі. Дано: Задачу про Ханойські башти. На башті А знаходиться `k` дисків різного діаметру. Кожен диск меншого діаметра лежить на диску більшого. Всього башт є три. Необхідно: перемістити `k` дисків з башти А на башту С, використовуючи як допоміжну башту В. За одне переміщення можна перемістити лише один диск. Диск більшого діаметра не можна класти на диск меншого діаметра. Створити програму засобами мови С, використавши для виконання переміщень рекурсивну функцію.

```
/* **** */
/* lab_work_15_01. */
/* Рекурсія. Задача про Ханойські башти. */
/* Дано: На башті А знаходиться k дисків різного діаметру. */
/* Кожен диск меншого діаметра лежить на диску більшого */
/* діаметра. Всього башт є три. */
/* Необхідно: */
/* Перемістити k дисків з башти А на башту С, */
/* використовуючи як допоміжну башту В. */
/* За одне переміщення можна перемістити лише один диск. */
/* Диск більшого діаметра не можна класти на диск меншого */
/* діаметра. */
/* Створити програму засобами мови С, використавши для */
/* виконання переміщень рекурсивну функцію. */
/* **** */
#include <stdio.h>
#include <conio.h>
#include <windows.h>

int k;
char A='A', B='B', C='C';
int p;

void HanoiTowers(int n, char *source, char *dest, char *tmp);

int main() {
    clrscr();
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    printf(" Введіть кількість дисків k: ");
    scanf("%d", &k);

    //Виклик рекурсивної функції
```

```

HanojTowers(k, &A, &C, &B);

getch();
return 0;
}

//-----
// Рекурсивна функція
void HanojTowers(int n, char *source, char *dest, char *tmp)
{
    if (n==1) { //Варіант 1: переміщення диска номер 1
        printf("\n Перемістіть диск номер %d ",n);
        printf(" з башти %c на башту %c", *source,*dest);
    }
    else { // Варіант 2: переміщення дисків з іншими номерами
        // Рекурсивний виклик функції:
        HanojTowers(n-1, source,tmp,dest);
        printf("\n Перемістіть диск номер %d ",n);
        printf(" з башти %c на башту %c",*source,*dest);
        // Рекурсивний виклик функції:
        HanojTowers(n-1, tmp, dest, source);
    }
}

```

Результати тестування програми:

Введіть кількість дисків k: 4

```

Перемістіть диск номер 1 з башти А на башту В
Перемістіть диск номер 2 з башти А на башту С
Перемістіть диск номер 1 з башти В на башту С
Перемістіть диск номер 3 з башти А на башту В
Перемістіть диск номер 1 з башти С на башту А
Перемістіть диск номер 2 з башти С на башту В
Перемістіть диск номер 1 з башти А на башту В
Перемістіть диск номер 4 з башти А на башту С
Перемістіть диск номер 1 з башти В на башту С
Перемістіть диск номер 2 з башти В на башту А
Перемістіть диск номер 1 з башти С на башту А
Перемістіть диск номер 3 з башти В на башту С
Перемістіть диск номер 1 з башти А на башту В
Перемістіть диск номер 2 з башти А на башту С
Перемістіть диск номер 1 з башти В на башту С

```

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

1. Напишіть рекурсивну функцію знаходження суми цифр натурального числа.
2. Написати рекурсивну функцію обчислення x^n , де $n > 0$.
3. Написати рекурсивну функцію обчислення чисел Фібоначі, що визначаються за формулою $fib_{n+1} = fib_n + fib_{n-1}$ і $fib_1 = 1, fib_0 = 0$.

4. Описати рекурсивну функцію $\text{pow}(x,n)$, яка обчислює величину x^n , згідно з формулою

$$\begin{aligned} x^n &= 1, \text{ при } n=0; \\ x^n &= 1/x^{-n}, \text{ при } n<0; \\ x^n &= x * x^{n-1}, \text{ при } n>0. \end{aligned}$$

5. Обчислити декілька значень функції Аккермана для невід'ємних чисел m та n :

$$A(m, n) = \begin{cases} m+1, & n=0 \\ A(m-1, 1), & n \neq 0, m=0 \\ A(m-1, A(m, n-1)), & n>0, m \geq 0 \end{cases}$$

6. Написати функцію $C(m,n)$ для обчислення біноміальних коефіцієнтів C_m^n використовуючи наступну формулу:

$$C_m^0 = C_m^m = 1, \quad C_m^n = C_{m-1}^n + C_{m-1}^{n-1}, \quad \text{при } 0 < n < m.$$

7. Обчислити значення виразу, використовуючи рекурсію:

$$\sqrt{6 + 2\sqrt{7 + 3\sqrt{8 + 4\sqrt{9 + \dots}}}}$$

8. Обчислити значення $x = \sqrt{a}$ використовуючи рекурентну формулу $x_n = \frac{1}{2}(x_{n-1} + a/x_{n-1})$, за початкове наближення взяти значення $x_0 = 0.5(1+a)$.

9. Обчислити $N!$, використовуючи рекурсивний алгоритм.

10. Обчислити значення функції $y(n) = \sqrt{1 + \sqrt{2 + \dots + \sqrt{n}}}$

11. Обчислити значення функції

$$y = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \dots$$

12. Обчислити $y = x^N$ використовуючи наступний алгоритм: $y = (x^{N/2})^2$, якщо N парне; $y = x \cdot y^{N-1}$, якщо N непарне.

$$S = \sum_{i=1}^n (i+1)^2 / i$$

13. Обчислити S , використовуючи рекурсію.

14. Знайти найменший спільний дільник двох натуральних чисел.

Рекурентна формула алгоритма Евкліда, запишеться :

$$HCD(a,b) = \begin{cases} a, \text{ якщо } a=b \\ HCD(a-b,b), \text{ якщо } a>b \\ HCD(b,b-a), \text{ якщо } b>a \end{cases}$$

15. Вивести на екран усі зростаючі послідовності довжиною k , елементами яких є натуральні числа від 1 до n . Передбачається, що k не перевищує n , інакше таких послідовностей не існує.

ЛАБОРАТОРНА РОБОТА №16: “РОБОТА З ОДНОВИМІРНИМИ МАСИВАМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Вивчити методи роботи з одновимірними масивами засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання оголошення одновимірних масивів, введення та виведення значень їх елементів, обчислення суми та добутку елементів, знаходження найбільшого та найменшого значень, табулювання функції, аргумент якої задано масивом.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_16.01.

```
/* **** */
/* Смысл імені масиву одновимірного масиву */
/* **** */
#include <stdio.h>
#include <conio.h>

int main()
{clrscr();
  char array[5]={'A','L','P','H','A'};

  printf("\n\n");
  printf("    array    = %p\n\n",array);

  printf("    array[0] = %p\n", &array[0]);
  printf("    array[1] = %p\n", &array[1]);
  printf("    array[2] = %p\n\n", &array[2]);

  printf("    &array    = %p\n", &array);
  printf("    &array[1] = %p\n", &array[1]);
  printf("    &array[2] = %p\n\n", &array[2]);

  printf("    *array    = %c\n", *array);
  printf("    *(array+1)= %c\n", *(array+1));
  printf("    *(array+2)= %c\n", *(array+2));

  getch();
  return 0;
}
```

Результати тестування програми:

```
array    = FFF0
array[0] = FFF0
array[1] = FFF1
array[2] = FFF2
```

```
&array    = FFF0
&array[1] = FFF1
&array[2] = FFF2
```

```

*array      = A
*(array+1)= L
*(array+2)= P

```

2. Ввести, відлагодити та протестувати програму під назвою lab_work_16.02.

```

/*****
/*
/*      Дано масив a(a[1],a[2],...,a[m], елементи якого є */
/*      цілого типу і різні за величиною, та масив      */
/*      b(a[1],b[2],...,b[n], елементи якого є дійсного типу /
/*      і різні за величиною.                               */
/*      Необхідно: ввести значення всіх елементів масиву  */
/*      a[m] та надрукувати їх засобами мови C;           */
/*      ввести значення всіх елементів масиву b[n] та     */
/*      надрукувати їх засобами мови C++.                 */
/*      Доступ до елементів масивів здійснити за допомогою*/
/*      вказівників.                                     */
/*
/*****
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

void main()
{clrscr();
  int i,m,n;
  int a[100];
  float b[150];

  // Ввід та вивід значень елементів масиву a засобами мови C:
  // Ввід кількості елементів m масиву a:
  printf("\n Ввести кількість елементів m масиву a: ");
  scanf("%d",&m);

  // Ввід елементів масиву a:
  for (i=0;i<m; i++)
  {printf(" Ввести a[%d]:",i);
    scanf("%d",&a[i]);
  }

  // Вивід елементів масиву a:
  printf("\n Заданий масив a:\n");
  for (i=0;i<m; i++)
    printf(" %d :",a[i]);
  printf("\n");

  // Ввід та вивід значень елементів масиву b засобами мови C++:
  // Ввід кількості елементів n:
  cout<<"\n Ввести кількість елементів n масиву b: ";
  cin>>n;

  // Ввід елементів масиву b:
  for (i=0;i<n; i++)

```

```

    {cout<<" Ввести b["<<i<<"]: ";
      cin>>*(b+i);
    }

    // Вивід елементів масиву b:
    cout<<"\n    "<<"Заданий масив b:\n";
    for (i=0;i<n; i++)
        cout<<"    "<<*(b+i);
    cout<<"\n";

    getch();
}

```

Результати тестування програми:

```

Ввести кількість елементів m масиву a: 5
Ввести a[0]:4
Ввести a[1]:7
Ввести a[2]:2
Ввести a[3]:-1
Ввести a[4]:3

```

```

Заданий масив a:
2442 : -546 : 19921 : 2442 : 4 :

```

```

Ввести кількість елементів n масиву b: 3
Ввести b[0]: 7
Ввести b[1]: 2
Ввести b[2]: 8

```

```

Заданий масив b:
7 2 8

```

3. Ввести, відлагодити та протестувати програму під назвою lab_work_16.03.

```

/*****
/*
/* Дано масив a(a[1],a[2],...,a[m], елементи якого є
/* цілого типу і різні за величиною, та масив
/* b(a[1],b[2],...,b[n], елементи якого є дійсного типу*/
/* і різні за величиною.
/* Необхідно: ввести значення всіх елементів масиву
/* a[m] та надрукувати їх засобами мови C; ввести
/* значення всіх елементів масиву b[n] та надрукувати
/* їх засобами мови C++.
/* Доступ до елементів масивів здійснити за допомогою*/
/* індексів.
/*
/*****
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

```

```

void main()

```



```

{clrscr();
 int i,m,n;
 int a[100];
 float b[150];

// Ввід та вивід значень елементів масиву a засобами мови C:
// Ввід кількості елементів m масиву a:
printf("\n Ввести кількість елементів m масиву a: ");
scanf("%d",&m);

// Ввід елементів масиву a:
for (i=0;i<m; i++)
{printf(" Ввести a[%d]:",i);
 scanf("%d",&a[i]);
}
// Вивід елементів масиву a:
printf("\n Заданий масив a:\n");
for (i=0;i<m; i++)
 printf(" %d ",a[i]);
printf("\n");

// Ввід та вивід значень елементів масиву b засобами мови C++:
// Ввід кількості елементів n:
cout<<"\n Ввести кількість елементів n масиву b: ";
cin>>n;
// Ввід елементів масиву b:
for (i=0;i<n; i++)
{ cout<<" Ввести b["<<i<<"]: ";
 cin>>b[i];
}
// Вивід елементів масиву b:
cout<<"\n "<<"Заданий масив b:\n";
for (i=0;i<n; i++)
 cout<<" "<<b[i];
cout<<"\n";

getch();
}

```

Результати тестування програми:

```

Ввести кількість елементів m масиву a: 5
Ввести a[0]:4
Ввести a[1]:7
Ввести a[2]:2
Ввести a[3]:-1
Ввести a[4]:3

```

```

Заданий масив a:
2442 : -546 : 19921 : 2442 : 4 :

```

```

Ввести кількість елементів n масиву b: 3
Ввести b[0]: 7

```

```
Ввести b[1]: 2
Ввести b[2]: 8

Заданий масив b:
7 2 8
```

4. Ввести, відлагодити та протестувати програму під назвою lab_work_16.04.

```

/*****/
/* Табулювання функції. Обчислити всі значення */
/* функції  $y = \sin(3x)^2$ , аргумент якої набуває */
/* значень {-5, 3, 13, 4.3, 2.8, 3.14, 0, 8, 9.3, 10.4}. */
/* Розв'язати цю задачу для випадку, коли масив */
/* ініціалізовано 10 значеннями. Доступ до елементів */
/* масивів здійснити за допомогою індексів */
/* */
/*****/
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>

void main()
{clrscr();
 int i,n;
 float y[100],x[100]={-5,3,13,4.3,2.8,3.14,0,8,9.3,10.4};

 //Формування та друк елементів масиву y:
 cout<<"\n\n Результати табулювання:\n";
 for (i=0;i<10; i++)
 {y[i]=pow(sin(3*x[i]),2);
 printf(" x[%d]=%+6.2f y[%d]=%+6.3f\n",i,x[i],i,y[i]);
 }

 getch();
}
```

Результати тестування програми:

```

x[0]= -5.00 y[0]=+0.423
x[1]= +3.00 y[1]=+0.170
x[2]=+13.00 y[2]=+0.929
x[3]= +4.30 y[3]=+0.107
x[4]= +2.80 y[4]=+0.730
x[5]= +3.14 y[5]=+0.000
x[6]= +0.00 y[6]=+0.000
x[7]= +8.00 y[7]=+0.820
x[8]= +9.30 y[8]=+0.134
x[9]=+10.40 y[9]=+0.046
```

5. Ввести, відлагодити та протестувати програму під назвою lab_work_16.05.

```

/*****/
/* Дано масив: a(a[1],a[2],...,a[n]).Всі елементи */
/* цілого типу і різні за величиною. */
/* Необхідно знайти найбільший парний елемент та */
/*****/
```

```

/* його номер. Якщо парних елементів у масиві      */
/* немає, надрукуйте відповідне повідомлення.      */
/*****/
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
void main()
{clrscr();
  int i,n,nom,max,ozn;
  int a[100];

  // Ввід кількості елементів n:
  cout<<"\n Ввести кількість елементів масиву n: ";
  cin>>n;

  // Ввід елементів масиву a:
  for (i=0;i<n; i++)
  {cout<<" Ввести a["<<i<<"]: ";
    cin>>a[i];
  }

  ozn=0;
  //Знаходження найбільшого парного елемента max та його номера
  nom:

  for (i=0;i<n; i++)
  {if ((ozn==0)&& (a[i]%2==0))
    {ozn=1;
     nom=i;
     max=a[i];
    }
    else
    if ((a[i]%2==0) && (a[i]>max))
    {nom=i;
     max=a[i];
    }
  }
  if (ozn==0)
  cout<<"\n Парних елементів у масиві а немає.\n";
  else
  {cout<<"\n Найбільший парний елемент max="<<max<<"\n";
   cout<<"\n Його номер у масиві      nom="<<nom<<"\n";
  }
  getch();
}

```

Результати тестування програми:

```

Ввести кількість елементів масиву n: 5
Ввести a[0]: 2
Ввести a[1]: 9
Ввести a[2]: 4
Ввести a[3]: 8

```

Ввести a[4]: 3

Найбільший парний елемент max=8
Його номер у масиві nom=3

Ввести кількість елементів масиву n: 5
Ввести a[0]: 3
Ввести a[1]: 7
Ввести a[2]: 1
Ввести a[3]: 9
Ввести a[4]: 5

Парних елементів у масиві a немає.

6. Ввести, відлагодити та протестувати програму під назвою lab_work_16.06.

```
/*
 *
 * Табулювання функції. Обчислити всі значення
 * функції  $y = \sin(3x)^2$ , аргумент якої набуває
 * значень елементів заданого одновимірного масиву.
 * Задачу розв'язати для загального випадку,
 * коли масив складається з n елементів.
 * а протестувати на масиві
 * {-5,3,13,4.3,2.8,3.14,0,8,9.3,10.4}.
 * Доступ до елементів масивів здійснити за
 * допомогою вказівників. Мова C++.
 * Ввід та вивід елементів масивів та процес
 * табулювання оформити у вигляді функцій.
 */
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
#include <math.h>
#include <windows.h>
#define N 100

void OneDimArrInp(float a[N], int *nn);
void OneDimArrOut(float a[N], int nn);
void OneDimArrTab(float a[N], float b[N], int nn);

int i,n;
float y[100],x[100]; //={-5,3,13,4.3,2.8,3.14,0,8,9.3,10.4};

void main() {
    clrscr();
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    //Ввід елементів масиву x:
    OneDimArrInp(x,&n);
```

```

//Друк елементів масиву x:
OneDimArrOut(x,n);

//Табулювання функції y=f(x):
OneDimArrTab(x,y, n);

}

// Функція введення значень одновимірного масиву:
void OneDimArrInp(float a[N], int *nn){
    cout<<"\n Ведіть кількість елементів одновимірного масиву: ";
    cin>>*nn;
    for (int i = 0; i < *nn; i++) {
        cout<<"\n Ведіть значення елемента a["<<i<<"]="";
        cin>>*(a+i);
    }
}

// Функція виведення значень одновимірного масиву на екран:
void OneDimArrOut(float a[N], int nn){
    cout<<"\n Одновимірний масив a:";
    cout<<"\n ";
    for (int i = 0; i < nn; i++)
        cout<<*(a+i)<<" ";
    cout<<"\n ";
    cout<<"\n Натисніть будь-яку клавішу, щоб продовжити...\n";
    getch();
}

// Функція табулювання і виведення на екран значень функції,
// аргумент якої задано одновимірним масивом:
void OneDimArrTab(float a[N], float b[N], int nn){
    cout<<"\n\n Результати табулювання:\n";
    setw(5);
    for (i=0;i<nn; i++) {
        *(b+i)=pow(sin(3*(*(a+i))),2);
        cout<<"\n
                                x["<<i<<"]="<<setw(5)<<*(a+i)<<"
y["<<i<<"]="<<*(b+i);
    }
    cout<<"\n\n Розв'язок задачі завершено!";
    getch();
}

```

Результати тестування програми:

```

Ведіть кількість елементів одновимірного масиву: 10
Ведіть значення елемента a[0]=-5
Ведіть значення елемента a[1]=3
Ведіть значення елемента a[2]=13
Ведіть значення елемента a[3]=4.3
Ведіть значення елемента a[4]=2.8
Ведіть значення елемента a[5]=3.14

```

Ведіть значення елемента $a[6]=0$
Ведіть значення елемента $a[7]=8$
Ведіть значення елемента $a[8]=9.3$
Ведіть значення елемента $a[9]=10.4$

Одновимірний масив а:
-5 3 13 4.3 2.8 3.14 0 8 9.3 10.4

Натисніть будь-яку клавішу, щоб продовжити...

Сума елементів одновимірного масиву:
48.94

Результати табулювання:

$x[0]=$	-5	$y[0]=$	20.6955
$x[1]=$	3	$y[1]=$	8.31205
$x[2]=$	13	$y[2]=$	45.4604
$x[3]=$	4.3	$y[3]=$	5.24832
$x[4]=$	2.8	$y[4]=$	35.7428
$x[5]=$	3.14	$y[5]=$	0.0011172
$x[6]=$	0	$y[6]=$	0
$x[7]=$	8	$y[7]=$	40.1343
$x[8]=$	9.3	$y[8]=$	6.54332
$x[9]=$	10.4	$y[9]=$	2.24653

Розв'язок задачі завершено!

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Створити, відлагодити та протестувати програму для табулювання заданої функції $y=f(x)$, аргумент якої пробігає множину значень заданого одновимірного масиву $a[n]$ у відповідності з індивідуальним варіантом. Протестувати програму за різними наборами даних, один з яких – заданий в умові.

Варіанти індивідуальних завдань:

1. $y = \sin^2 x + S$, де S - сума елементів масиву $a[n]=[3, 0, -1, \pi, 2]$.
2. $y = \ln x + \sqrt{S}$, де S - добуток елементів масиву $a[n]=[1, 4, 3, 2, 6, 5]$.
3. $y = \cos^2 x + S$, де S - сума додатних елементів масиву $a[n]=[0, 2, -\pi, 1, -2]$.
4. $y = \sqrt[3]{\ln(x+5) - S}$, де S - сума від'ємних елементів масиву $a[n]=[0, -3, 2, -4, -1]$.
5. $y = \sqrt{\ln x + x + S}$, де S - добуток додатних елементів масиву $a[n]=[1, 2, 4, 3, 5]$.
6. $y = \arctg(x)/(\cos^2 x + S)$, де S - сума елементів масиву $a[n]$, більших за число 2; $a[n]=[-1, 0, \pi, 3, -2]$.
7. $y = ((x-5)^2 + \sin^3 x)/S$, де S - середнє арифметичне елементів масиву $a[n]=[1, 0, 4, 6, 2, 3]$.
8. $y = \sqrt{x-5} + (4+S)/3$, де S - сума елементів масиву $a[n]$, кратних числу 3; $a[n]=[5, 6, 9, 6, 12, 7]$.
9. $y = \sin^2 x + S/x$, де S - середнє геометричне елементів масиву $a[n]=[1, 3, 2, 5, 4, 6]$.

10. $y = (S + \sqrt{x+4})^2$, де S - добуток від'ємних елементів масиву $a[n]=[-1, -3, 0, 5, -2]$.
11. $y = S + \sqrt[3]{\cos^2(x+5)}$, де S - кількість додатних елементів масиву $a[n]=[-5, 2, 0, -1, 1]$.
12. $y = \arctg(x^3 + 3) - S$, де S - сума елементів масиву з парними індексами; $a[n]=[-1, 2, 3, -4, 0]$.
13. $y = S + \cos x + e^x$, де S - сума елементів масиву з непарними індексами; $a[n]=[0, 1, -2, \pi, 3]$.
14. $y = S - \ln \sqrt{e^x + \sin^2 x}$, де S - кількість нульових елементів масиву; $a[n]=[0, 1, 5, 0, -2]$.
15. $y = \cos^2 x + S\sqrt{e^x}$, де $S=1$, - якщо всі елементи масиву $a[n]$ додатні, та $S=0$, - в протилежному випадку; $a[n]=[-1, 1, 2, 3, \pi]$.
16. $y = e^{(5+(\cos x)/\ln x)} + S$, де S - сума елементів масиву $a[n]=[1, 2, 3, 4, 5]$.
17. $y = (\cos^2 x + \arctg^2(x+5))/(S+x)$, де S - добуток елементів масиву $a[n]=[-1, 2, 4, -2, 1]$.
18. $y = \sqrt[3]{x^2+1} + S$, де S - сума додатних елементів масиву $a[n]=[-2, 0, 3, 1, 1]$.
19. $y = S \ln(x^2/3)$, де S - сума від'ємних елементів масиву $a[n]=[-3, -2, 4, 0, -1]$.
20. $y = \text{ctg}(S/5) + \sqrt{x^2+4}$, де S - добуток додатних елементів масиву $a[n]=[0, -2, 1, -1, 2]$.
21. $y = \arctg(3\pi + x + S)$, де S - сума елементів масиву $a[n]$, більших за число 3; $a[n]=[4, 3, 2, 5, 0, 1]$.
22. $y = x^2 + \ln(S + (x-2)^2)$, де S - середнє арифметичне значення елементів масиву $a[n]=[2, 0, 1, 4, 3]$.
23. $y = Sx/(x^2 + \sqrt{x+2})$, де S - середнє арифметичне значення додатних елементів масиву $a[n]=[-1, 0, -2, 1, 2, 3]$.
24. $y = (1 + \cos^2 x)(S + x^3)$, де S - сума елементів масиву $a[n]$, кратних числу 2; $a[n]=[2, 1, 0, 4, 3]$.
25. $y = S + \sqrt{|x|+2}$, де S - середнє геометричне значення елементів масиву $a[n]=[0, 1, -1, 2, 3]$.
26. $y = \sin x + \sqrt{|S|}$, де S - добуток від'ємних елементів масиву $a[n]=[-1, 0, -2, 1, -5]$.
27. $y = S - \sqrt[3]{x+1}$, де S - кількість додатних елементів масиву $a[n]=[1, 2, 0, -1, 4]$.
28. $y = \sin^2 2x + S$, де S - сума елементів масиву з парними індексами; $a[n]=[2, 0, -2, 1, 3]$.
29. $y = \cos^2(x+S)$, де S - кількість нульових елементів масиву $a[n]=[0, 1, 0, 4, 0]$.
30. $y = e^{x+1}(2x^2 + S)$, де $S=0$, - якщо всі елементи масиву $a[n]$ додатні, та $S=1$, - в протилежному випадку; $a[n]=[1, 2, 3, -1, 4]$.

Завдання 2. Створити, відлагодити та протестувати програму для обробки заданого одновимірного масиву у відповідності з індивідуальним варіантом.

1. Визначити, яких елементів в даному масиві $a[a[1], a[2], \dots, a[n]]$ найбільше: від'ємних, додатних чи нульових.
2. Перетворити заданий масив $a[a[1], a[2], \dots, a[n]]$ так, щоб в ньому спочатку були розташовані від'ємні елементи, потім додатні, а потім нульові.
3. Визначити, який елемент заданого масиву $a[a[1], a[2], \dots, a[n]]$ стоїть ближче до початку - найбільший чи найменший.
4. Визначити, скільки елементів заданого масиву $a[a[1], a[2], \dots, a[n]]$ стоїть між найбільшим та найменшим елементами цього масиву, при умові, що вони єдині.

5. Поміняти місцями в заданому масиві $a(a[1], a[2], \dots, a[n])$ найбільший та найменший елементи.
6. Визначити, скільки в заданому масиві $a(a[1], a[2], \dots, a[n])$ елементів, кратних числу 5.
7. Обчислити суму парних елементів в заданому масиві $a(a[1], a[2], \dots, a[n])$ та їх кількість.
8. Обчислити суму непарних елементів в заданому масиві $a(a[1], a[2], \dots, a[n])$ та їх кількість.
9. Переставити елементи масиву $a(a[1], a[2], \dots, a[n])$ в зворотньому порядку
10. Замінити кожен від'ємний елемент масиву $a(a[1], a[2], \dots, a[n])$ нулем, а кожен нульовий елемент масиву – числом -1
11. Перетворити масив $a(a[1], a[2], \dots, a[8])$ за таким законом: перший елемент масиву піднести до степеня 8, другий - до 7 і т.д., восьмий-до 1 степеня.
12. Перетворити масив $a(a[1], a[2], \dots, a[n])$ за таким законом:
 $a[i] = a[1] + a[2] + \dots + a[i], i = 1, 2, 3, \dots, n.$
13. Визначити яких елементів в масиві $a(a[1], a[2], \dots, a[n])$ більше, парних чи непарних.
14. Визначити суму елементів масиву $a(a[1], a[2], \dots, a[n])$, які належать інтервалу (1, 5),
15. Зменшити вдвічі кожен елемент масиву $a(a[1], a[2], \dots, a[n])$, який більший за число 5.
16. Перетворити масив $a(a[1], a[2], \dots, a[n])$ за таким законом: $a[i] = a[i]/i, i = 1, 2, 3, \dots, n.$
17. Створити масив $a(a[1], a[2], \dots, a[10])$ елементи якого обчислюються за формулою:
 $a[i] = i!, i = 1, 2, 3, \dots, 10.$
18. Обчислити суми 1-го та 2-го, 3-го та 4-го, $p-1$ -го та n -го елементів масиву $a(a[1], a[2], \dots, a[n])$, якщо відомо, що n - число парне.
19. Поміняти місцями елементи масиву $a(a[1], a[2], \dots, a[n])$ за таким законом: 1-й на місце 2-го, 2-й, на місце 3-го, і т.д., n -й на місце 1-го.
20. Збільшити кожен другий елемент масиву $a(a[1], a[2], \dots, a[n])$ на число, яке дорівнює половині попереднього елемента цього ж масиву.
21. Зменшити кожен третій елемент масиву $a(a[1], a[2], \dots, a[n])$ на число, яке дорівнює половині першого сусіднього зліва елемента масиву.
22. Створити масив $a(a[1], a[2], \dots, a[8])$ елементи якого обчислюються за формулою:
 $a[i] = 1/(i+2), i = 1, 2, 3, \dots, 8.$
23. Замінити кожен нульовий елемент масиву $a(a[1], a[2], \dots, a[n])$, середнім арифметичним всіх елементів масиву,
24. Знайти суму тих елементів масиву $a(a[1], a[2], \dots, a[n])$, для яких $\sin(a[i]) > 0, i = 1, 2, \dots, n.$
25. Який елемент заданого масиву $a(a[1], a[2], \dots, a[n])$ розміщений ближче до кінця - найбільший чи найменший,
26. Знайти номер останнього елемента масиву $a(a[1], a[2], \dots, a[n])$, який кратний числам 3 та 5 одночасно.
27. Знайти номер першого елемента масиву $a(a[1], a[2], \dots, a[n])$, який кратний числам 2 та 7 одночасно.
28. Знайти суму елементів масиву $a(a[1], a[2], \dots, a[n])$, які стоять між найбільшим та найменшим елементами, при умові, що вони єдині.
29. Замінити кожен нульовий елемент масиву $a(a[1], a[2], \dots, a[n])$, середнім геометричним модулів елементів масиву відмінних від 0.
30. Знайти суму першої половини масиву $a(a[1], a[2], \dots, a[n])$ і суму решти його елементів. Число n вважати парним.

ЛАБОРАТОРНА РОБОТА №17: “ВПОРЯДКУВАННЯ ОДНОВИМІРНИХ МАСИВІВ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Вивчити методи впорядкування одновимірних масивів засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання впорядкування одновимірних масивів методами: мінімального елемента; обмінів; вставок; рекурсії та інші. У процесі самостійного складання програм за варіантом завдання рекомендується дотримуватися таких правил: програми складати мовою C++; максимально використовувати розроблені самостійно функції; звертання до елементів масивів організовувати за допомогою вказівників.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_17.01.

```
/*
*****
/*
/* Перестановка елементів масиву у зворотному порядку. */
/* Дано одновимірний масив a, що містить n елементів. */
/* Переставити елементи масиву у зворотному порядку. */
/* Доступ до елементів масиву здійснити за допомогою */
/* вказівників. */
/*
*****
#include <iostream.h>
#include <conio.h>
#include <math.h>

void main()
{clrscr();
 int i,j,m,n;
 float a[100];
 float b;

 // Ввід кількості елементів масиву:
 cout<<"\n Ввести кількість елементів масиву n: ";
 cin>>n;

 // Ввід елементів масиву a:
 for (i=0;i<n; i++)
 {cout<<" Ввести a["<i<<"]=";
  cin>>*(a+i);
 }

 m=n/2; // визначення середини масиву

 // Перестановка елементів масиву у зворотному порядку:
 j=n;
 for (i=0; i<m; i++)
 {b=*(a+i);
  *(a+i)=*(a+j-1);
  *(a+j-1)=b;
  j--;
```

```

    }
    // Друк елементів перетвореного масиву a:
    cout<<"Перетворений масив a:";
    for (i=0; i<n; i++) cout<<" "<<*(a+i)<<" ";

    getch();
}

```

Результати тестування програми:

```

Ввести кількість елементів масиву n: 7
Ввести a[0]=1
Ввести a[1]=6
Ввести a[2]=2
Ввести a[3]=8
Ввести a[4]=3
Ввести a[5]=4
Ввести a[6]=5
Перетворений масив a:  5   4   3   8   2   6   1

```

2. Ввести, відлагодити та протестувати програму під назвою lab_work_17.02.

```

/*****
/*
/* Впорядкування елементів масиву методом мінімального
/* елемента.
/* Дано одномірний масив a, що містить n елементів.
/* Переставити елементи масиву у порядку зростання
/* значень. Доступ до елементів масиву здійснити за
/* допомогою індексів.
/*
*****/
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#define SIZE 100
int main()
{clrscr();
  int a[SIZE];
  int min; // мінімальний елемент невідсортованої частини масиву
  int nom; // номер мінімального елемента масиву
  int k;   // номер кроку методу
  int i,n;

  // Ввід кількості елементів масиву:
  cout<<"\n"<<"Ввести кількість елементів масиву n<="<<SIZE<<": ";
  cin>>n;

  // Ввід елементів заданого масиву a:
  for (i=0; i<n; i++)
  {cout<<" Ввести a["<<i<<"]=";
    cin>>a[i];
  }
}

```

```
// Друк елементів заданого масиву а:
cout<<"\n"<<"  Заданий масив до впорядкування:"<<"\n";
for (i=0; i<n; i++)
    cout<<" "<<a[i]<<" ";
cout<<"\n";

// Впорядкування масиву методом мінімального елемента:
for (k=0; k<n-1; k++)
{min = a[k];
 nom = k;
 for (i=k+1; i<n; i++)
     if (a[i]<min)
     {
         min=a[i];
         nom=i;
     }
 a[nom]=a[k];
 a[k]=min;
}

// Друк елементів перетвореного масиву а:
cout<<"\n"<<"  Заданий масив після впорядкування:"<<"\n";
for (i=0; i<n; i++) cout<<" "<<a[i]<<" ";
cout<<"\n";

getch();
return 0;
}
```

Ввести кількість елементів масиву $n \leq 100$: 5
Ввести $a[0]=7$
Ввести $a[1]=2$
Ввести $a[2]=-1$
Ввести $a[3]=3$
Ввести $a[4]=4$

Заданий масив до впорядкування:
7 2 -1 3 4

Заданий масив після впорядкування:
-1 2 3 4 7

3. Ввести, відлагодити та протестувати програму під назвою lab_work_17.03.

```

/*****
/*
/*  Впорядкування елементів масиву методом обмінів.
/*  Дано одномірний масив а, що містить n елементів.
/*  Переставити елементи масиву у порядку зростання
/*  значень. Доступ до елементів масиву здійснити за
/*  допомогою індексів.
/*
*****/
#include <stdio.h>
```

```

#include <iostream.h>
#include <conio.h>
#define SIZE 100
int main()
{clrscr();
 int a[SIZE];
 int i,n,h,p;

 // Ввід кількості елементів масиву:
 cout<<"\n"<<"Ввести кількість елементів масиву n<="<<SIZE<<": ";
 cin>>n;

 // Ввід елементів масиву a:
 for (i=0; i<n; i++)
 {cout<<" Ввести a["<<i<<"]="";
  cin>>a[i];
 }

 // Друк елементів заданого масиву a:
 cout<<"\n"<<" Заданий масив до впорядкування:"<<"\n";
 for (i=0; i<n; i++) cout<<" "<<a[i]<<" ";
 cout<<"\n";
 // Впорядкування масиву a методом обмінів:
 for (p=1; p<n; p++)
 {for (i=0; i<n-1; i++)
  if (a[i]>a[i+1])
  {
   h=a[i];
   a[i]=a[i+1];
   a[i+1]=h;
  }
 }

 // Друк елементів перетвореного масиву a:
 cout<<"\n"<<" Заданий масив після впорядкування:"<<"\n";
 for (i=0; i<n; i++) cout<<" "<<a[i]<<" ";
 cout<<"\n";

 getch();
 return 0;
}

```

Результати тестування програми:

```

Ввести кількість елементів масиву n<=100: 5
Ввести a[0]=7
Ввести a[1]=2
Ввести a[2]=-1
Ввести a[3]=3
Ввести a[4]=4

```

Заданий масив до впорядкування:

```
7  2  -1  3  4
```

Заданий масив після впорядкування:
-1 2 3 4 7

4. Ввести, відлагодити та протестувати програму іпід назвою lab_work_17.04.

```
/*
 * Впорядкування елементів масиву методом вставки.
 * Дано одномірний масив a, що містить n елементів.
 * Переставити елементи масиву у порядку зростання
 * значень. Доступ до елементів масиву здійснити за
 * допомогою індексів.
 */
#include <stdio.h>
#include <iostream.h>
#include <conio.h>
#define SIZE 100

int main()
{
    clrscr();
    int a[SIZE];
    int b;
    int i,j,k,n;

    // Ввід кількості елементів масиву:
    cout<<"\n"<<"Ввести кількість елементів масиву n<="<<SIZE<<": ";
    cin>>n;

    // Ввід елементів масиву a:
    for (i=0; i<n; i++)
    {
        cout<<" Ввести a["<<i<<"]="";
        cin>>a[i];
    }

    // Друк елементів заданого масиву a:
    cout<<"\n"<<" Заданий масив до впорядкування:"<<"\n";
    for (i=0; i<n; i++) cout<<" "<<a[i]<<" ";
    cout<<"\n";

    // Впорядкування масиву методом вставки:
    for (k=1; k<n; k++)
    {
        b=a[k];
        j=0;
        while ((b>a[j]) && (j<k)) j++;
        if (j<k)
        {
            b=a[k];
            for(i=k; i>=j+1; i--) a[i]=a[i-1];
            a[j] = b;
        }
    }
}
```

```

// Друк елементів перетвореного масиву a:
cout<<"\n"<<" Заданий масив після впорядкування:"<<"\n";
for (i=0; i<n; i++)
    cout<<" "<<a[i]<<" ";
cout<<"\n";

getch();
return 0;
}

```

Результати тестування програми:

```

Ввести кількість елементів масиву n<=100: 5
Ввести a[0]=7
Ввести a[1]=2
Ввести a[2]=-1
Ввести a[3]=3
Ввести a[4]=4

```

```

Заданий масив до впорядкування:
7  2  -1  3  4

```

```

Заданий масив після впорядкування:
-1  2  3  4  7

```

5. Ввести, відлагодити та протестувати програму під назвою lab_work_17.05.

```

/*****
/*
/* Впорядкування елементів масиву методом рекурсії.
/* Дано одномірний масив a, що містить n елементів.
/* Переставити елементи масиву у порядку зростання
/* значень. Доступ до елементів масиву здійснити за
/* допомогою індексів.
/*
/*
*****/
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#define SIZE 100

int QuickSort(int l, int r, int mas[]);

int main()
{clrscr();
  int a[SIZE];
  int i,n;

  // Ввід кількості елементів масиву:
  cout<<"\n"<<"Ввести кількість елементів масиву n<="<<SIZE<<": ";
  cin>>n;

  // Ввід елементів масиву a:

```

```

for (i=0; i<n; i++)
{cout<<"  Ввести a["<<i<<"]=";
  cin>>a[i];
}

// Друк елементів заданого масиву a:
cout<<"\n"<<"  Заданий масив до впорядкування:"<<"\n";
for (i=0; i<n; i++)    cout<<" "<<a[i]<<" ";
cout<<"\n";

// Впорядкування масиву методом рекурсії:
QuickSort(0,n-1,a);

// Друк елементів перетвореного масиву a:
cout<<"\n"<<"  Заданий масив після впорядкування:"<<"\n";
for (i=0; i<n; i++) cout<<" "<<a[i]<<" ";
cout<<"\n";

getch();
return 0;
}

//-----
int QuickSort(int l, int r, int mas[])
{int i,j,b,tmp;
  b=mas[(l+r)/2];
  i=l;
  j=r;
  while (i<=j)
  {while (mas[i]<b) i++;
    while (mas[j]>b) j--;
    if (i<=j)
    {if (i!=j)
      {printf( "\n  l=%4d    i=%4d yes j=%4d    r=%d",l,i,j,r);
        tmp=mas[i];
        mas[i]=mas[j];
        mas[j]=tmp;
      }
      i++;
      j--;
    }
  }
  if (l<j)
  {QuickSort(l,j,mas);
    printf( "\n  l=%4d    i=%4d        j=%4d    r=%d",l,i,j,r);
  };

  if (i<r)
  {QuickSort(i,r,mas);
    printf( "\n  l=%4d    i=%4d        j=%4d    r=%d",l,i,j,r);
  };
  return 1;
}

```

Результати тестування програми:

Ввести кількість елементів масиву $n \leq 100$: 5
Ввести $a[0]=7$
Ввести $a[1]=2$
Ввести $a[2]=-1$
Ввести $a[3]=3$
Ввести $a[4]=4$

Заданий масив до впорядкування:
7 2 -1 3 4

Заданий масив після впорядкування:
-1 2 3 4 7

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Створити, відлагодити та протестувати програму для розв'язування такої задачі. Дано два одномірні масиви $a(a[1], a[2], \dots, a[m])$ та $b(b[1], b[2], \dots, b[n])$, розміри яких рівні m та n відповідно. Впорядкувати їх вказаними методами. Впорядковані масиви злити в масив $c[c[1], c[2], \dots, c[m+n]]$.

№ варіанту	Розмір m масиву a	Розмір n масиву b	Метод впорядкування масиву a	Метод впорядкування масиву b
1	2	3	4	5
1.	10	15	метод обміну	метод вставки
2.	8	12	метод вставки	метод вибору
3.	14	9	метод вибору	метод обміну
4.	7	11	метод вставки	метод обміну
5.	13	6	метод вибору	метод вставки
6.	15	12	метод обміну	метод вибору
7.	11	7	метод обміну	метод вставки
8.	6	10	метод вставки	метод вибору
9.	9	13	метод вибору	метод обміну
10.	15	8	метод вставки	метод обміну
11.	5	7	метод вибору	метод вставки
12.	8	14	метод обміну	метод вибору
13.	7	5	метод обміну	метод вставки
14.	16	9	метод вставки	метод вибору
15.	6	12	метод вибору	метод обміну
16.	9	5	метод вставки	метод обміну
17.	4	11	метод вибору	метод вставки
18.	8	6	метод обміну	метод вибору
19.	9	4	метод обміну	метод вставки
20.	7	9	метод вставки	метод вибору
21.	11	6	метод вибору	метод обміну
22.	4	15	метод вставки	метод обміну
23.	10	12	метод вибору	метод вставки
24.	9	8	метод обміну	метод вибору
25.	5	15	метод обміну	метод вставки
26.	13	4	метод вставки	метод вибору
27.	9	13	метод вибору	метод обміну
28.	6	4	метод вставки	метод обміну
29.	7	10	метод вибору	метод вставки
30.	8	5	метод обміну	метод вибору

ЛАБОРАТОРНА РОБОТА №18: “РОБОТА З ДВОВИМІРНИМИ МАСИВАМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Вивчити методи роботи з двовимірними масивами засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання оголошення двоовимірних масивів, введення та виведення значень їх елементів, обчислення суми та добутку елементів, знаходження найбільшого та найменшого значень, та інші.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_18_01.

```
/*
 *
 *      Смысл імені двовимірного масиву
 *
 */
/*****/
#include <stdio.h>
#include <conio.h>

int main()
{clrscr();
  char a[2][3]={'Q','W','E','R','T','Y'};

  // Використання імені масиву без індексів:
  // Друк адреси елемента a[0][0]
  printf("\n\n");
  printf(" Використання імені масиву без індексів:\n");
  printf("  a          = %p\n",a);
  // Друк значення елемента a[0][0]
  printf(" Імя масиву a без індексів - це адреса елемента
        a[0][0]\n\n",a);
  getch();

  // Використання посилань на елементи масиву:
  printf(" Використання посилань на елементи масиву:\n");
  // Друк адреси елемента a[0][0]
  printf("  &a          = %p\n", &a);
  // Друк адреси елемента a[0][0]
  printf("  &a[0][0]    = %p\n", &a[0][0]);
  // Друк адреси елемента a[0][1]
  printf("  &a[0][1]    = %p\n", &a[0][1]);
  // Друк адреси елемента a[0][2]
  printf("  &a[0][2]    = %p\n", &a[0][2]);
  // Друк адреси елемента a[1][0]
  printf("  &a[1][0]    = %p\n", &a[1][0]);
  // Друк адреси елемента a[1][1]
  printf("  &a[1][1]    = %p\n", &a[1][1]);
  // Друк адреси елемента a[1][2]
  printf("  &a[1][2]    = %p\n", &a[1][2]);
  printf(" Посилання на елемент масиву &a[i][j] - це адреса
        елемента a[i][j]\n\n");
```

```

    getch();

// Використання вказівників на елементи масиву:
printf(" Використання вказівників на елементи масиву:\n");
// Друк значення елемента a[0][0]
printf("    **a          = %c\n", **a);
// Друк значення елемента a[0][0]
printf("    *(* (a+0)+0)= %c\n", *(* (a+0)+0));
// Друк значення елемента a[0][1]
printf("    *(* (a+0)+1)= %c\n", *(* (a+0)+1));
// Друк значення елемента a[0][2]
printf("    *(* (a+0)+2)= %c\n", *(* (a+0)+2));
// Друк значення елемента a[1][0]
printf("    *(* (a+1)+0)= %c\n", *(* (a+1)+0));
// Друк значення елемента a[1][1]
printf("    *(* (a+1)+1)= %c\n", *(* (a+1)+1));
// Друк значення елемента a[1][2]
printf("    *(* (a+1)+2)= %c\n", *(* (a+1)+2));
printf(" Вказівник на елемент масиву *(* (a+i)+j) - це значення
        елемента a[i][j]\n\n");
getch();

// Використання індексів елементів масиву:
printf(" Використання індексів елементів масиву:\n");
// Друк значення елемента a[0][0]
printf("    a[0][0]= %c\n", a[0][0]);
// Друк значення елемента a[0][1]
printf("    a[0][1]= %c\n", a[0][1]);
// Друк значення елемента a[0][2]:
printf("    a[0][2]= %c\n", a[0][2]);
// Друк значення елемента a[1][0]
printf("    a[1][0]= %c\n", a[1][0]);
// Друк значення елемента a[1][1]
printf("    a[1][1]= %c\n", a[1][1]);
// Друк значення елемента a[1][2]
printf("    a[1][2]= %c\n", a[1][2]);
printf(" Елемент масиву a[i][j] - це значення елемента
        a[i][j]\n\n");
getch();
return 0;
}

```

Результати тестування програми:

Використання імені масиву без індексів:

a = FFF0

Ім'я масиву a без індексів - це адреса елемента a[0][0]

Використання посилань на елементи масиву:

&a = FFF0

&a[0][0] = FFF0

&a[0][1] = FFF1

&a[0][2] = FFF2

```

&a[1][0]    = FFF3
&a[1][1]    = FFF4
&a[1][2]    = FFF5

```

Посилання на елемент масиву &a[i][j] - це адреса елемента a[i][j]

Використання вказівників на елементи масиву:

```

**a          = Q
*(*(a+0)+0) = Q
*(*(a+0)+1) = W
*(*(a+0)+2) = E
*(*(a+1)+0) = R
*(*(a+1)+1) = T
*(*(a+1)+2) = Y

```

Вказівник на елемент масиву *(*(a+i)+j) - це значення елемента a[i][j]

```

&a[1][0]    = FFF3
&a[1][1]    = FFF4
&a[1][2]    = FFF5

```

Посилання на елемент масиву &a[i][j] - це адреса елемента a[i][j]

Використання вказівників на елементи масиву:

```

**a          = Q
*(*(a+0)+0) = Q
*(*(a+0)+1) = W
*(*(a+0)+2) = E
*(*(a+1)+0) = R
*(*(a+1)+1) = T
*(*(a+1)+2) = Y

```

Вказівник на елемент масиву *(*(a+i)+j) - це значення елемента a[i][j]

Використання індексів елементів масиву:

```

a[0][0] = Q
a[0][1] = W
a[0][2] = E
a[1][0] = R
a[1][1] = T
a[1][2] = Y

```

Елемент масиву a[i][j] - це значення елемента a[i][j]

2. Ввести, відлагодити та протестувати програму під назвою lab_work_18_02.

```

/*****
/*
/*      Дано масив a[m][n], всі елементи якого є цілого
/*      типу і різні за величиною, та масив b[p][q], всі
/*      елементи якого є дійсного типу і різні за величиною.
/*      Необхідно ввести значення всіх елементів масиву
/*      a[m][n] та надрукувати їх засобами мови C; ввести
/*      значення всіх елементів масиву b[p][q] та
/*      надрукувати їх засобами мови C++.
/*      Доступ до елементів масивів здійснити за допомогою
/*      індексів.
*/

```

```

/*                                                                 */
/*****
#include <iostream.h>
#include <stdio.h>
#include <stdio.h>
#include <conio.h>

void main()
{clrscr();
  int i,j,m,n,p,q;
  int a[10][10];
  float b[15][15];
  // Ввід та вивід значень елементів масиву a засобами мови C:
  // Ввід кількості елементів m масиву a:
  printf("\n  Ввести кількість стрічок m масиву a: ");
  scanf("%d",&m);
  printf("  Ввести кількість стовпчиків n масиву a: ");
  scanf("%d",&n);

  // Ввід елементів масиву a:
  for (i=0;i<m; i++)
    for (j=0; j<n;j++)
      {printf("  Ввести a[%d][%d]:",i,j);
        scanf("%d",&a[i][j]);
      }

  // Вивід елементів масиву a:
  printf("\n  Заданий масив a:\n");
  for (i=0;i<m; i++)
    {for (j=0; j<n;j++)
      printf("  %d  ",a[i][j]);
      printf("\n");
    }

  // Ввід та вивід значень елементів масиву b засобами мови C++:
  // Ввід кількості стрічок p та стовпчиків q масиву b:
  cout<<"\n  Ввести кількість стрічок p масиву b: ";
  cin>>p;
  cout<<"  Ввести кількість стовпчиків q масиву b: ";
  cin>>q;
  // Ввід елементів масиву b:
  for (i=0;i<p; i++)
    {for (j=0;j<q;j++)
      {cout<<"  b["<<i<<"] ["<<j<<"]: ";
        cin>>b[i][j];
      }
    }
  // Вивід елементів масиву b:
  cout<<"\n  "<<"Заданий масив b:\n";
  for (i=0;i<p; i++)
    {for (j=0;j<q;j++)
      cout<<"  "<<b[i][j]<<"  ";
      cout<<"\n";
    }
}

```

```

    }
    getch();
}

```

Результати тестування програми:

```

Ввести кількість стрічок m масиву a: 2
Ввести кількість стовпчиків n масиву a: 3
Ввести a[0][0]:1
Ввести a[0][1]:2
Ввести a[0][2]:3
Ввести a[1][0]:4
Ввести a[1][1]:5
Ввести a[1][2]:6

```

```

Заданий масив a:
1      2      3
4      5      6

```

```

Ввести кількість стрічок p масиву b: 2
Ввести кількість стовпчиків q масиву b: 2
b[0][0]: 1
b[0][1]: 2
b[1][0]: 3
b[1][1]: 4

```

```

Заданий масив b:
1      2
3      4

```

3. Ввести, відлагодити та протестувати програму під назвою lab_work_18_03.

```

/*****
/*
/*      Дано двовимірний масив a[m][n], елементи
/*      якого дійсного типу (m стрічок, n стовпців).
/*      Необхідно обчислити суми елементів по
/*      стрічках у вигляді масиву g[m] та суми
/*      елементів по стовпцях у вигляді масиву g[m].
/*      Результат вивести у вигляді матриці.
/*      Доступ до елементів масивів здійснити за
/*      допомогою індексів.
/*
*****/
#include <stdio.h>
#include <conio.h>
#include <iostream.h>

void main()
{clrscr();
  float a[10][10];
  float g[10];
  float v[10];
  float s,d;

```

```

int m,n;
int i,j;

// Введення числа стрічок та стовпців:
cout<<"\n Введіть к-сть стрічок масиву: ";
cin>>m;
cout<<" Введіть к-сть стовпців масиву: ";
cin>>n;

// Введення значень двовимірного масиву:
cout<<"\n Введіть поелементно заданий масив:\n";

for (i=0;i<m; i++)
    for (j=0;j<n; j++)
        {cout<<" Введіть a["<<i<<"]["<<j<<"]="";
          cin>>a[i][j];
        }

// Друк заданого масиву:
cout<<"\n\n Заданий масив:\n";
for (i=0;i<m; i++)
    {for (j=0;j<n;j++)
        cout<<" "<<a[i][j];
      cout<<"\n";
    }

// Обчислення сум кожного рядка:
for (i=0;i<m; i++)
    {g[i]=0;
      for (j=0; j<n; j++) g[i]+=a[i][j];
    }

// Обчислення сум кожного стовпця:
for (j=0;j<n; j++)
    {v[j]=0;
      for (i=0;i<m;i++) v[j]+=a[i][j];
    }

// Друк заданого масиву та сум по рядках та стовпцях:
cout<<"\n\n Заданий масив і суми по рядках і стовпцях:\n";
for (i=0;i<m; i++)
    {for (j=0;j<n;j++) cout<<" "<<a[i][j];
      cout<<" "<<g[i]<<"\n";
    }
for (j=0;j<n; j++) cout<<" "<<v[j];

getch();
}

```

Результати тестування програми:

Введіть к-сть стрічок масиву: 2
Введіть к-сть стовпців масиву: 3

Введіть поелементно заданий масив:

```
Введіть a[0][0]=1
Введіть a[0][1]=2
Введіть a[0][2]=3
Введіть a[1][0]=4
Введіть a[1][1]=5
Введіть a[1][2]=6
```

Заданий масив:

```
1  2  3
4  5  6
```

Заданий масив і суми по рядках і стовпцях:

```
1  2  3  6
4  5  6  15
5  7  9
```

4. Ввести, відлагодити та протестувати програму під назвою lab_work_18_04.

```
/*
*****/
/*
Дано двовимірний масив a[m][n], елементи
якого дійсного типу (m стрічок, n стовпців).
Необхідно знайти мінімальний та максимальний
елементи масиву та значення їх індексів.
Доступ до елементів масивів здійснити за
допомогою індексів.
*/
*****/
```

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
void main()
{clrscr();
 float a[10][10];
 float min, max;

 int m,n;
 int i,j;
 int nomstr, nomsto;
 int numstr, numsto;

 // Введення числа стрічок та стовпців:
 cout<<"\n Введіть к-сть стрічок масиву: ";
 cin>>m;

 cout<<" Введіть к-сть стовпців масиву: ";
 cin>>n;

 // Введення значень двовимірного масиву:
```

```

cout<<"\n Введіть поелементно заданий масив:\n";
for (i=0;i<m; i++)
    for (j=0;j<n; j++)
        {cout<<" Введіть a["<<i<<"] ["<<j<<"]=";
          cin>>a[i][j];
        }

// Друк заданого масиву:
cout<<"\n\n Заданий масив:\n";
for (i=0;i<m; i++)
    {for (j=0;j<n;j++)
      cout<<" "<<a[i][j];
      cout<<"\n";
    }

// Обчислення мінімального та максимального значень масиву
// та їх індексів:
min=a[0][0];
nomstr=0;
nomsto=0;

max=a[0][0];
numstr=0;
numsto=0;
for (i=0;i<m; i++)
    for (j=0;j<n;j++)
        {if (a[i][j]<min)
          {min=a[i][j];
           nomstr=i;
           nomsto=j;
          }
          if (a[i][j]>max)
          {max=a[i][j];
           numstr=i;
           numsto=j;
          }
        }

// Друк мінімального та максимального значень масиву:
cout<<"\n Мінімальний елемент масиву min="<<min<<"\n";
cout<<"\n Номер стічки мінімального елемента масиву
nomstr="<<nomstr<<"\n";
cout<<"\n Номер стовця мінімального елемента масиву
nomsto="<<nomsto<<"\n";
cout<<"\n Максимальний елемент масиву max="<<max<<"\n";
cout<<"\n Номер стічки максимального елемента масиву
numstr="<<numstr<<"\n";
cout<<"\n Номер стовця максимального елемента масиву
numsto="<<numsto<<"\n";

getch();
}

```


Результати тестування програми:

Введіть к-сть стрічок масиву: 2
Введіть к-сть стовпців масиву: 3

Введіть поелементно заданий масив:

Введіть $a[0][0]=5$
Введіть $a[0][1]=8$
Введіть $a[0][2]=2$
Введіть $a[1][0]=9$
Введіть $a[1][1]=1$
Введіть $a[1][2]=3$

Заданий масив:

5	8	2
9	1	3

Мінімальний елемент масиву $\min=1$

Номер стічки мінімального елемента масиву $\text{nomstr}=1$

Номер стовця мінімального елемента масиву $\text{nomsto}=1$

Максимальний елемент масиву $\max=9$

Номер стічки максимального елемента масиву $\text{numstr}=1$

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Дано двовимірний масив $a[m][n]$ елементів (m стрічок, n стовпчиків). Створити, відлагодити та протестувати програму для розв'язування задачі, заданої нижче.

1. Знайти суму додатніх елементів кожної стрічки.
2. Знайти суму від'ємних елементів кожного стовпчика.
3. Знайти середнє арифметичне додатніх елементів кожного стовпчика.
4. Знайти середнє арифметичне від'ємних елементів кожної стрічки.
5. Знайти суму елементів, які перевищують по модулю одиницю, в кожному парному стовпчику.
6. Знайти суму елементів, які не перевищують π , в кожній непарній стрічці.
7. Знайти суму квадратів елементів, які розміщені на діагоналях ($m=n$).
8. Знайти середнє арифметичне додатніх елементів і середнє арифметичне від'ємних елементів.
9. Знайти добуток ненульових елементів.
10. Знайти добуток додатніх елементів і добуток від'ємних елементів.
11. Знайти добуток ненульових елементів кожної непарної стрічки.
12. Знайти добуток додатніх елементів масиву, що розміщені в місцях перетину парних стрічок і непарних стовпчиків.
13. Знайти суму елементів кожної стрічки і суму елементів кожного стовпчика.
14. Знайти значення максимального елемента кожного стовпчика.
15. Знайти значення мінімального по модулю елемента кожної стрічки.
16. Знайти значення мінімального та максимального елементів.
17. Знайти значення максимального елемента кожної стрічки і номер стовпчика, в якому він розміщений,
18. Знайти значення мінімального елемента кожного стовпчика і номер стрічки, в якій він розміщений.
19. Знайти значення максимального елемента і його індекси.
20. Знайти середнє геометричне додатніх елементів кожного стовпчика.

21. Знайти середнє гармонійне додатніх елементів кожної стрічки.
22. Знайти середнє геометричне додатних елементів, які розміщені в місцях перетину непарних стрічок і парних стовпчиків.
23. Знайти кількість додатніх і кількість від'ємних елементів.
24. Знайти кількість додатніх елементів кожної стрічки і всього масиву.
25. Знайти кількість від'ємних елементів у всій матриці і в кожному стовпчику.
26. Знайти кількість і номери стрічок, які включають нульові елементи.
27. Знайти для кожного парного стовпчика кількість нульових елементів і їх добуток.
28. Знайти для кожного непарного рядка кількість ненульових елементів і суму їх добутків.
29. Вивести на екран новий масив отриманий з початкового діленням всіх елементів на максимальний по модулю елемент.
30. Вивести на екран новий масив, отриманий з початкового множенням елементів кожного рядка на мінімальний елемент цього рядка.

ЛАБОРАТОРНА РОБОТА №19: “ФОРМУВАННЯ ДВООВИМІРНИХ МАСИВІВ ЗА ЗАДАНИМ ЗАКОНОМ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Вивчити методи формування двовимірних масивів за заданим законом засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

знаходження матриці, яка є добутком двох заданих матриць;

формування двовимірного масиву з елементів двовимірного масиву за заданим законом.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_19_01.

```
/*
*****/
/*
Дано матрицю (двовимірний масив) a[m][n]
(m стрічок, n стовпців) та матрицю
(двовимірний масив) b[p][q] (p стрічок, q
стовпців). Елементи обох матриць дійсного
типу .
Необхідно обчислити елементи матриці c=a*b,
яка є добутком двох заданих матриць.
Результат вивести у вигляді матриці.
Доступ до елементів масивів здійснити за
допомогою вказівників.
*/
*****/
#include <stdio.h>
#include <conio.h>
#include <iostream.h>

void main()
{clrscr();
float a[10][10];
float b[10][10];
float c[10][10];
float s,d;
int m,n,p,q;
int i,j,k;

// Введення числа стрічок та стовпців матриці a:
cout<<"\n Введіть к-сть стрічок матриці a: ";
cin>>m;
cout<<" Введіть к-сть стовпців матриці a: ";
cin>>n;
// Введення числа стрічок та стовпців матриці b:
cout<<"\n Введіть к-сть стрічок матриці b: ";
cin>>p;
cout<<" Введіть к-сть стовпців матриці b: ";
cin>>q;

if (n==p) cout<<"\n Множення матриць можливе.\n";
```

```

else
{cout<<"\n  Множення матриць не можливе.";
  cout<<"\n  Перевірте розміри матриць! \n";
}

// Введення значень матриці a:
cout<<"\n  Введіть поелементно матрицю a:\n";
for (i=0;i<m; i++)
  for (j=0;j<n; j++)
    {cout<<"  Введіть a["<<i<<"] ["<<j<<"]=";
      cin>>* (a+i)+j);
    }

// Введення значень матриці b:
cout<<"\n  Введіть поелементно матрицю b:\n";
for (i=0;i<p; i++)
  for (j=0;j<q; j++)
    {cout<<"  Введіть a["<<i<<"] ["<<j<<"]=";
      cin>>* (b+i)+j);
    }

// Друк матриці a:
cout<<"\n\n  Матриця a:\n";
for (i=0;i<m; i++)
  {for (j=0;j<n;j++)
    cout<<"    "<<* (a+i)+j);
    cout<<"\n";
  }

// Друк матриці b:
cout<<"\n\n  Матриця b:\n";
for (i=0;i<p; i++)
  {for (j=0;j<q;j++)
    cout<<"    "<<* (b+i)+j);
    cout<<"\n";
  }

// Обчислення елементів матриці c:
for (i=0;i<m; i++)
  for (j=0;j<q;j++)
    {* (* (c+i)+j)=0;
      for (k=0;k<n;k++)
        * (* (c+i)+j) += * (* (a+i)+k) * (* (* (b+k)+j));
    }

// Друк матриці c:
cout<<"\n\n  Матриця-добуток c:\n";
for (i=0;i<m; i++)
  {for (j=0;j<q;j++)
    cout<<"    "<<* (* (c+i)+j);
    cout<<"\n";
  }

```

```

    }
    getch();
}

```

Результати тестування програми:

Введіть к-сть стрічок матриці a: 2
Введіть к-сть стовпців матриці a: 2

Введіть к-сть стрічок матриці b: 2
Введіть к-сть стовпців матриці b: 3

Множення матриць можливе.

Введіть поелементно матрицю a:
Введіть a[0][0]=1
Введіть a[0][1]=2
Введіть a[1][0]=3
Введіть a[1][1]=4

Введіть поелементно матрицю b:
Введіть a[0][0]=1
Введіть a[0][1]=2
Введіть a[0][2]=3
Введіть a[1][0]=4
Введіть a[1][1]=5
Введіть a[1][2]=6
Введіть a[1][1]=4

Матриця a:
1 2
3 4

Матриця b:
1 2 3
4 5 6

Матриця-добуток c:
9 12 15
19 26 33

2. Ввести, відлагодити та протестувати програму під назвою lab_work_19_02.

```

/*****
/* Дано натуральні числа n, m=n*(n+1)/2. */
/* Дано одновимірний масив a[m], елементи */
/* якого дійсного типу. */
/* Необхідно сформувати і надрукувати */
/* двовимірний масив b[n][n], розташовуючи у */
/* ньому елементи одномірного масиву за */
/* наведеною на малюнку схемою. */
/* Результат вивести у вигляді матриці. */

```

```

/*      Доступ до елементів масивів здійснити за      */
/*      допомогою індексів.                          */
/*                                                    */
/*****/
#include <stdio.h>
#include <conio.h>
#include <iostream.h>

void main()
{clrscr();
 float a[100];
 float b[10][10];
 int m,n;
 int i,j,k;

 // Введення числа елементів масиву n:
 cout<<"\n Введіть натуральне число n: ";
 cin>>n;
 // Визначення кількості елементів одновимірного масиву a:
 m=n*(n+1)/2;
 // Введення значень одновимірного масиву:
 cout<<"\n Введіть поелементно заданий масив:\n";
 for (i=0;i<m; i++)
 {cout<<" Введіть a["<i<<"]=";
  cin>>a[i];
 }

 // Друк заданого масиву:
 cout<<"\n\n Заданий масив:\n";
 for (i=0;i<m; i++)
 {cout<<" "<a[i];
 }
 cout<<"\n";
 // Обнулення елементів двовимірного масиву:
 for (i=0;i<n; i++)
 for (j=0;j<n;j++) b[i][j]=0;

 // Формування двовимірного масиву:
 k=0;
 for (j=0; j<n; j++)
 if ((j+2)%2>0)
 // Формування непарного стовпчика:
 for (i=n-1; i>-1; i--)
 if (i<j)
 b[i][j]=0;
 else
 {
 b[i][j]=a[k];
 k++;
 }
 else
 // Формування парного стовпчика:
 for (i=0; i<n; i++)

```

```

        if (i<j)
            b[i][j]=0;
        else
            {b[i][j]=a[k];
              k++;
            };

// Друк сформованого за схемою двовимірного масиву:
cout<<"\n\n  Сформований двовимірний масив:\n";
for (i=0;i<n; i++)
    {for (j=0;j<n;j++) cout<<"    "<<b[i][j];
      cout<<"\n";
    }

getch();
}

```

Результати тестування програми:

Введіть натуральне число n: 4

Введіть поелементно заданий масив:

Введіть a[0]=1

Введіть a[1]=2

Введіть a[2]=3

Введіть a[3]=4

Введіть a[4]=5

Введіть a[5]=6

Введіть a[6]=7

Введіть a[7]=8

Введіть a[8]=9

Введіть a[9]=1

Заданий масив:

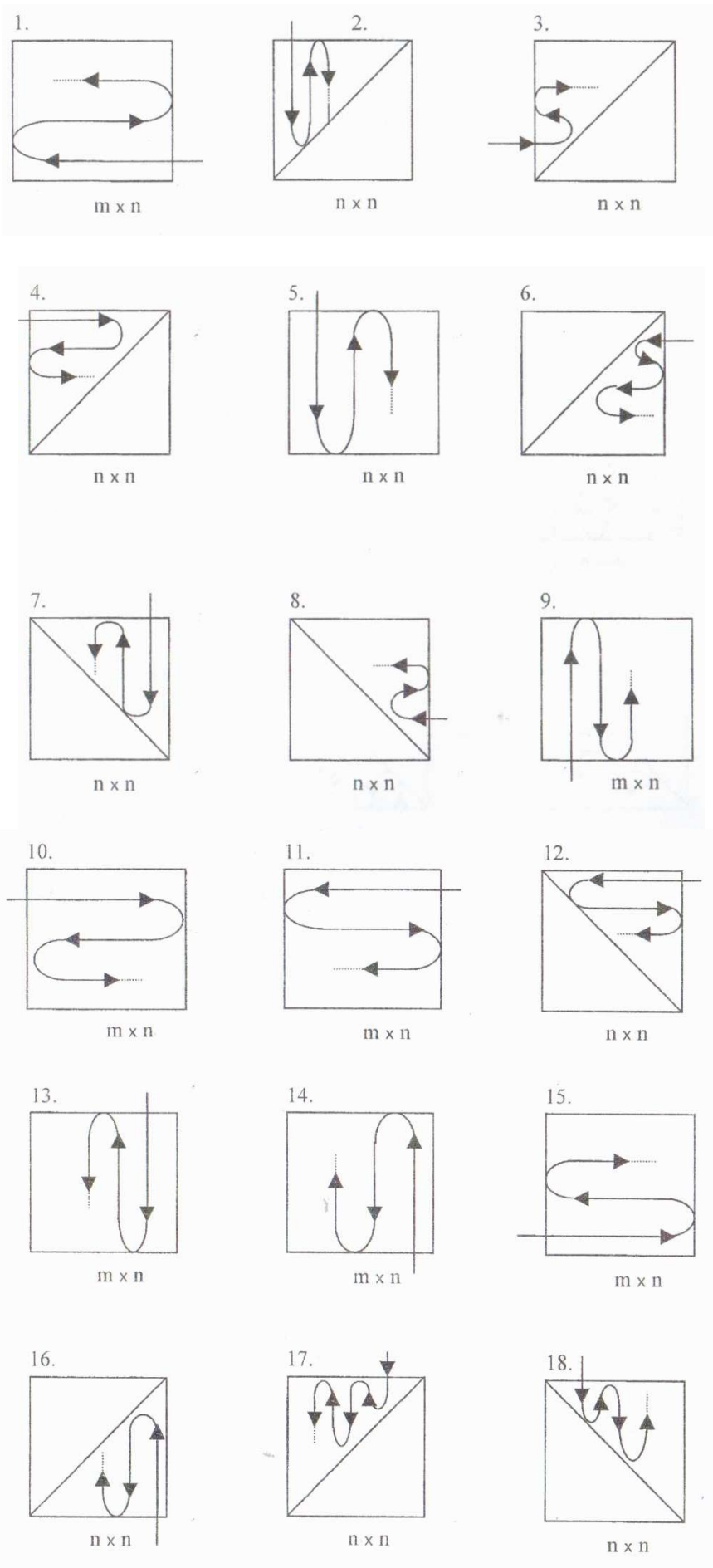
1	2	3	4	5	6	7	8	9	1
---	---	---	---	---	---	---	---	---	---

Сформований двовимірний масив:

1	0	0	0
2	7	0	0
3	6	8	0
4	5	9	1

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Створити, відлагодити та протестувати програму для формування двовимірного масиву $a[m][n]$ шляхом зчитування в нього даних зі заданого одновимірного масиву $b[m*n]$ згідно з індивідуальним варіантом.



ЛАБОРАТОРНА РОБОТА №20: “РОБОТА З ФУНКЦІЯМИ ОБРОБКИ СИМВОЛІВ ТА СТРІЧОК ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику використання функцій з бібліотек обробки символів та стрічок засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

робота з функціями з бібліотеки обробки символів <ctype.h>;

робота з функціями з бібліотеки обробки стрічок <string.h>;

робота з функціями зі стандартної бібліотеки введення/виведення <stdio.h>;

робота з функціями зі стандартної бібліотеки загального призначення <stdlib.h>.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_20_01.

```
/*
 *
 * Робота з функціями для обробки символів з бібліотеки
 * обробки символів: <ctype.h>:
 *      int isdigit(int c); int isalpha(int c);
 *      int isalnum(int c); int isxdigit(int c).
 *
 */
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
main()
{clrscr();
 // int isdigit(int c) - повертає істинне значення, якщо c - ціла
 // цифра, і 0 - в протилежному випадку
 printf("%s\n%s\n\n",
        " Демонструємо роботу з функцією isdigit: ",
        isdigit('8') ? " 8 є ": " 8 не є ", "цифрою",
        isdigit('#') ? " # є ": " # не є ", "цифрою" );

 // int isalpha(int c) - повертає істинне значення, якщо c -
 // буква, і 0 - в протилежному випадку
 printf("%s\n%s\n\n",
        " Демонструємо роботу з функцією isalpha:",
        isalpha('A') ? " A є ": " A не є ", "буквою",
        isalpha('b') ? " b є ": " b не є ", "буквою",
        isalpha('&') ? " & є ": " & не є ", "буквою",
        isalpha('4') ? " 4 є ": " 4 не є ", "буквою");

 // int isalnum(int c) - повертає істинне значення, якщо c - ціла
 // цифра або буква, і 0 - в протилежному випадку
 printf("%s\n%s\n\n",
        " Демонструємо роботу з функцією isalnum:",
        isalnum('A') ? " A є ": " A не є ", "цифрою або буквою",
        isalnum('8') ? " 8 є ": " 8 не є ", "цифрою або буквою",
        isalnum('#') ? " # є ": " # не є ", "цифрою або буквою" );
```

```
// int isxdigit(int c) ) -повертає істинне значення, якщо c -
// шістнадцяткова цифра, і 0 - в протилежному випадку
printf("%s\n%s%s\n%s%s\n%s%s\n%s%s\n%s%s\n\n",
      " Демонструємо роботу з функцією isxdigit:",
isxdigit('F') ? " F є ": " F не є ", "шістнадцятковою цифрою",
isxdigit('J') ? " J є ": " J не є ", "шістнадцятковою цифрою",
isxdigit('7') ? " 7 є ": " 7 не є ", "шістнадцятковою цифрою",
isxdigit('$') ? " $ є ": " $ не є ", "шістнадцятковою цифрою",
isxdigit('f') ? " f є ": " f не є ", "шістнадцятковою цифрою"
getch();
return 0;
}
```

Результати тестування програми:

Демонструємо роботу з функцією **isdigit**:

8 є цифрою
не є цифрою

Демонструємо роботу з функцією **isalpha**:

A є буквою
b є буквою
& не є буквою
4 не є буквою

Демонструємо роботу з функцією **isalnum**:

A є цифрою або буквою
8 є цифрою або буквою
не є цифрою або буквою

Демонструємо роботу з функцією **isxdigit**:

F є шістнадцятковою цифрою
J не є шістнадцятковою цифрою
7 є шістнадцятковою цифрою
\$ не є шістнадцятковою цифрою
f є шістнадцятковою цифрою

2. Ввести, відлагодити та протестувати програму під назвою lab_work_20_02.

```
/*
 *
 * Робота з функціями для обробки символів з бібліотеки
 * обробки символів: <ctype.h>;
 *      int islower(int c); int isupper(int c);
 *      int tolower(int c); int toupper(int c);
 *
 */
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
main()
{ clrscr();
  // int islower(int c) - функція повертає 1 (істинне значення),
  // якщо c - буква нижнього регістру, і 0 - в протилежному випадку
```

```

printf("%s\n%s%s\n%s%s\n%s%s\n\n",
    " Демонструємо роботу з функцією islower:",
    islower('p') ? " р є ": " р не є ", "буква нижнього регістру",
    islower('P') ? " Р є ": " Р не є ", "буква нижнього регістру",
    islower('5') ? " 5 є ": " 5 не є ", "буква нижнього регістру",
    islower('!') ? " ! є ": " ! не є ", "буква нижнього регістру");

// int isupper(int c) - функція повертає 1 (істинне значення),
// якщо c - буква верхнього регістру, і 0 - в протилежному
//випадку
printf("%s\n%s%s\n%s%s\n%s%s\n\n",
    " Демонструємо роботу з функцією isupper:",
    isupper('p') ? " р є ": " р не є ", "буква верхнього регістру",
    isupper('P') ? " Р є ": " Р не є ", "буква верхнього регістру",
    isupper('5') ? " 5 є ": " 5 не є ", "буква верхнього регістру",
    isupper('!') ? " ! є ": " ! не є ", "буква верхнього регістру");

// int toupper(int c)- якщо c - буква нижнього регістру,
// то функція повертає значення c у верхньому регістрі,
// в протилежному випадку значення c повертається без змін
printf("%s\n%s%c\n%s%c\n%s%c\n%s%c\n\n",
    " Демонструємо роботу з функцією toupper:",
    " Символ р у верхньому регістрі буде ", toupper('p'),
    " Символ 7 у верхньому регістрі буде ", toupper('7'),
    " Символ $ у верхньому регістрі буде ", toupper('$'),
    " Символ L у верхньому регістрі буде ", tolower('L'));
getch();
return 0;
}

```

Результати тестування програми:

Демонструємо роботу з функцією **islower**:

```

р є буква нижнього регістру
Р не є буква нижнього регістру
5 не є буква нижнього регістру
! не є буква нижнього регістру

```

Демонструємо роботу з функцією **isupper**:

```

р не є буква верхнього регістру
Р є буква верхнього регістру
5 не є буква верхнього регістру
! не є буква верхнього регістру

```

Демонструємо роботу з функцією **toupper**:

```

Символ р у верхньому регістрі буде Р
Символ 7 у верхньому регістрі буде 7
Символ $ у верхньому регістрі буде $
Символ L у верхньому регістрі буде l

```

3. Ввести, відлагодити та протестувати програму під назвою lab_work_20_03.

```

/*****
/*
/*  Робота з функціями для обробки символів з бібліотеки */

```

```

/*  обробки символів: <ctype.h>;                                     */
/*      int isspace(int c); int iscntrl(int c);                       */
/*      int ispunct(int c); int isgraph(int c);                       */
/*                                                                 */
/*****/
#include <stdio.h>
#include <ctype.h>
#include <conio.h>
main()
{clrscr();
  // Функція int isspace(int c) повертає істинне значення,
  // якщо c - пробільний символ ( \n \t \f \r \v),
  // та 0 -у протилежному випадку
  printf("%s\n%s%s\n%s%s\n%s%s\n\n",
    " Демонструємо роботу з функцією isspace:",
    "  \n", isspace('\n') ? " є ": "не є ",
    "білопробільним символом",
    "  \t", isspace('\t') ? " є ": "не є ",
    "білопробільним символом",
    "  % ", isspace('%') ? " є ": "не є ",
    "білопробільним символом");

  // Функція int iscntrl(int c) повертає істинне значення,
  // якщо c - є управляючим символом,
  // та 0 - у протилежному випадку
  printf("%s\n%s%s\n%s%s\n%s%s\n\n",
    " Демонструємо роботу з функцією iscntrl:",
    "  \n", iscntrl('\n') ? " є ": " не є ",
    "управляючим символом",
    "  \t", iscntrl('\t') ? " є ": " не є ",
    "управляючим символом",
    "  % ", iscntrl('%') ? " є ": " не є ",
    "управляючим символом");

  // Функція ispunct(int c) повертає істинне значення,
  // якщо c - є друкованим символом, що не є буквою,
  // цифрою або пробілом, та 0 - у протилежному випадку
  printf("%s\n%s%s\n%s%s\n%s%s\n\n",
    " Демонструємо роботу з функцією ispunct:",
    "  ;", ispunct(';') ? " є ": " не є ",
    "друкованим символом, що не є буквою, цифрою або пробілом",
    "  Y", ispunct('Y') ? " є ": " не є ",
    "друкованим символом, що не є буквою, цифрою або пробілом",
    "  #", ispunct('#') ? " є ": " не є ",
    "друкованим символом, що не є буквою, цифрою або пробілом");

  // Функція int isprint(int c) повертає істинне значення,
  // якщо c - є друкованим символом або пробілом,
  // та 0 -у протилежному випадку
  printf("%s\n%s%s\n%s%s\n%s%s\n\n",
    " Демонструємо роботу з функцією isprint:",
    "  $", isprint('$') ? " є ": " не є ",
    "друкованим символом або пробілом",

```

```

"  \\a", isprint('\\a') ? " є ": " не є ",
"друкованим символом або пробілом",
"  A", isprint('A') ? " є ": " не є ",
"друкованим символом або пробілом");

// Функція int isgraph(int c) повертає істинне значення,
// якщо c - є друкованим символом, але не пробілом,
// та 0 -у протилежному випадку
printf("%s\\n%s%s%s\\n%s%s%s\\n%s%s%s\\n\\n",
" Демонструємо роботу з функцією isgraph:",
"  $", isgraph('$') ? " є ": " не є ",
"друкованим символом, але не пробілом",
"  _", isgraph('_') ? " є ": " не є ",
"друкованим символом, але не пробілом",
"  A", isgraph('A') ? " є ": " не є ",
"друкованим символом, але не пробілом");
getch();
return 0;
}

```

Результати тестування програми:

Демонструємо роботу з функцією **isspace**:

```

\\n є білопробільним символом
\\t є білопробільним символом
% не є білопробільним символом

```

Демонструємо роботу з функцією **iscntrl**:

```

\\n є управляючим символом
\\t є управляючим символом
% не є управляючим символом

```

Демонструємо роботу з функцією **ispunct**:

```

; є друкованим символом, що не є буквою, цифрою або пробілом
Y не є друкованим символом, що не є буквою, цифрою або пробілом
# є друкованим символом, що не є буквою, цифрою або пробілом

```

Демонструємо роботу з функцією **isprint**:

```

$ є друкованим символом або пробілом
\\a не є друкованим символом або пробілом
A є друкованим символом або пробілом

```

Демонструємо роботу з функцією **isgraph**:

```

$ є друкованим символом, але не пробілом
_ не є друкованим символом, але не пробілом
A є друкованим символом, але не пробілом

```

4. Ввести, відлагодити та протестувати програму під назвою `lab_work_20_04`.

```

/*****
/*
/* Функція для перетворення стрічок з бібліотеки
/* утиліт загального призначення <stdlib.h>:
/* double atof(const char *nPtr);
*/

```

```

/*      int atoi(const char *nPtr);      */
/*      long atol(const char *nPtr);      */
/*      */
/*****/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int main()
{clrscr();
 // Функція double atof(const char *nPtr) перетворює
 // стрічку nPtr в число типу double:
 // Змінна для запам'ятовування результату перетворення:
 double d;
 d=atof("99.0");

 printf("%s%.3f\n%s%.3f\n\n",
 " Стрічка \"99.0\", перетворена у число типу double: ",d,
 " Отримане число, розділене на 2, дорівнює: ", d/2.0);

 // Функція int atoi(const char *nPtr) перетворює
 // стрічку nPtr в число типу int:

 // Змінна для запам'ятовування результату перетворення:
 int i;
 i=atoi("2593");

 printf("%s%d\n%s%d\n\n",
 " Стрічка \"2593\", перетворена у число типу int: ",i,
 " Отримане число мінус 593, дорівнює: ", i-593);

 // Функція double atol(const char *nPtr) перетворює
 // стрічку nPtr в число типу double:

 // Змінна для запам'ятовування результату перетворення:
 long l;
 l=atol("1000000");

 printf("%s%ld\n%s%ld\n\n",
 " Стрічка \"1000000\", перетворена у число типу long: ",l,
 " Отримане число, розділене на 2, дорівнює: ", l/2);
 getch();
 return 0;
}

```

Результати тестування програми:

Стрічка "99.0", перетворена у число типу double: 99.000
Отримане число, розділене на 2, дорівнює: 49.500

Стрічка "2593", перетворена у число типу int: 2593
Отримане число мінус 593, дорівнює: 2000

Стрічка "1000000", перетворена у число типу long: 1000000

Отримане число, розділене на 2, дорівнює: 500000

5. Ввести, відлагодити та протестувати програму під назвою lab_work_20_05.

```
/*
 *
 * Функції для введення/виведення стрічок зі
 * стандартної бібліотеки введення/виведення <stdio.h>:
 *
 * char *gets(char *s);
 * int puts(const char *s);
 * int putchar(int c);
 *
 */
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
void reverse( const char * const sPtr);
int main()
{
    clrscr();
    char s[20]; // Задана стрічка з 20 символів
    int i, len;

    // Функція char *gets(char *s) вводить символи зі
    // стандартного пристрою вводу/виводу у символьний масив s
    // до тих пір, поки не зустрине символ нової стрічки
    // або індикатор кінця файлу. Після цього до масиву s
    // додає завершальний нуль
    printf(" Введіть стрічку і натисніть клавішу Ввід:\n ");
    gets(s);
    len=strlen(s); // len - довжина стрічки s

    // Функція int putchar(const char *s) друкує один символ
    // зі стрічки s
    printf("\n Посимвольний друк стрічки s:\n ");
    for (i=0; i<len; i++) putchar(s[i]);

    // Функція int puts(const char *s) друкує всю стрічку s
    // відразу і переводить курсор на початок нової стрічки
    printf("\n\n%s\n%s\n%s", " Друк всієї стрічки s відразу",
           " і перевод курсора на початок нової стрічки:", " ");
    puts(s);

    printf("\n Друк стрічки у зворотному порядку:\n ");
    reverse(s);
    getch();
    return 0;
}
//-----
//Інверсія стрічки
void reverse( const char * const sPtr)
{
    if (sPtr[0] == '\0') return;
    else
        {reverse(&sPtr[1]);

```



```

    putchar(sPtr[0]);
}
}

```

Результати тестування програми:

Введіть стрічку і натисніть клавішу Ввід:
 Луцьк - обласний центр Волині

Посимвольний друк стрічки s:
 Луцьк - обласний центр Волині

Друк всієї стрічки s відразу
 і перевод курсора на початок нової стрічки:
 Луцьк - обласний центр Волині

Друк стрічки у зворотньому порядку:
 інилоВ ртнец йинсалбо - кьцуЛ

6. Ввести, відлагодити та протестувати програму під назвою lab_work_20_06.

```

/*****
/*
/*      Функції введення/виведення стрічок зі стандартної
/*      бібліотеки введення/виведення <stdio.h>:
/*      int getchar(void);
/*      int sprintf(const char *s, const char *format, ...);
/*      int sscanf(char *s, const char *format, ...);
/*
/*****
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
int main()
{clrscr();
  char c;
  char s[20]; // Задана стрічка з 20 символів
  int i=0;
  puts(" Введіть стрічку з 20 (або менше) символів:");
  printf(" ");

// Функція int getchar(void) вводить один символ зі
// стандартного пристрою введення/виведення і повертає
// його у формі цілого
while ((c=getchar()) != '\n') s[i++]=c;
s[i]='\0'; // В кінці додаємо завершальний нуль
printf("\n Введена посимвольно стрічка:\n ");
puts(s);

  int x;
  double y;
  printf("\n Введіть два числа - типу int та типу double:\n ");
  scanf("%d%lf", &x,&y);

```

```
// Функція int sprintf(char *s, const char *format, ...)  
// еквівалентна функції printf, однак вивід здійснюється  
// в масив s, а не відображується на екрані  
sprintf(s, "   ціле: %6d\n   дійсне:%8.2f",x,y);  
printf("\n%s\n%s\n",  
      "   Форматований вигляд вмісту масиву s є таким:",s);  
  
char s1[]="31298 87.375";  
// Функція int sscanf(char *s, const char *format, ...)  
// еквівалентна функції scanf, однак ввід здійснюється  
// з масиву s, а не з клавіатури  
sscanf(s1, "%d%lf", &x,&y);  
printf("\n%s\n%s%6d\n%s%8.3f\n",  
      "   Величини, записані в масиві s1, є такими:",  
      "   ціле:", x, "   з плаваючою комою:", y);  
getch();  
return 0;  
}
```

Результати тестування програми:

Введіть стрічку з 20 (або менше) символів:
Луцьк - столиця Волині

Введена посимвольно стрічка:
Луцьк - столиця Волині

Введіть два числа - типу int та типу double:
34 45.56

Форматований вигляд вмісту масиву `s` є таким:

```
ціле:      34
дійсне:    45.56
```

Величини, записані в масиві s1, є такими:
ціле: 31298
з плаваючою комою: 87.375

7. Ввести, відлагодити та протестувати програму під назвою lab work 20 07.

```

/*****
/*
/*    Функції для перетворення стрічок з бібліотеки
/*    утиліт загального призначення <stdlib.h>:
/*
/*    double strtod(const char *nPtr,char **endPtr);
/*
/*    long strtol(const char *nPtr, char **endPtr, int base);
/*
/*    unsigned long strtoul(const char *nPtr, char **endPtr,
/*    int base);
/*
*****/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main()
{clrscr();
  // Функція double strtod(const char *nPtr, char **endPtr)
  // перетворює стрічку nPtr в число типу double:
  const char *string1 = "51.2 % від кількості учнів";
  char *stringPtr; // Вказівник на char

  // Змінна для запам'ятовування результату перетворення:
  double d;
  d=strtod(string1, &stringPtr);

  printf("  Стрічка \"%s\" \n конвертується на дві частини:\n",
    string1);

  printf("  число %.2f типу double;\n  та стрічку: \" %s \" \n",
    d, stringPtr);

  // Функція long strtol(const char *nPtr, char **endPtr,
  // int base)
  // перетворює стрічку nPtr в число типу long:
  const char *string2 = "-1234567abc";
  char *rPtr; // Вказівник на char

  // Змінна для запам'ятовування результату перетворення:
  long x;
  x=strtol(string2, &rPtr,0);

  printf("\n  Стрічка \"%s\" \n конвертується на дві частини:\n",
    string2);

  printf("  число %ld типу long;\n  та стрічку: \" %s \" \n", x,
    rPtr);

  // Функція unsigned long strtoul(const char *nPtr, char
  // **endPtr,int base)
  // перетворює стрічку nPtr в число типу unsigned long:
  const char *string3 = "1234567abc";
  char *qPtr; // Вказівник на char
  // Змінна для запам'ятовування результату перетворення:
  unsigned long y;
  y=strtol(string3, &qPtr,0);
  printf("\n  Стрічка \"%s\" \n конвертується на дві частини:\n",
    string3);
  printf("  число %lu типу unsigned long;\n  та стрічку: \" %s \" \n",y,qPtr);
  getch();
  return 0;
}

```

Результати тестування програми:

Стрічка "51.2 % від кількості учнів"
конвертується на дві частини:
число 51.20 типу double;
та стрічку: " % від кількості учнів "

Стрічка "-1234567abc"
конвертується на дві частини:
число -1234567 типу long;
та стрічку: " abc "

Стрічка "1234567abc"
конвертується на дві частини:
число 1234567 типу unsigned long;
та стрічку: " abc "

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Створити, відлагодити та протестувати програму для обробки заданої стрічки (стрічок) згідно з індивідуальним варіантом.

1. Текст задано слідуєчим чином: перший символ - цифра, що вказує довжину слова, за словом - цифра, що вказує довжину другого слова і т.д. Довжина кожного слова не перевищує 9. Надрукувати к- те слово.
2. Задано текст, слова якого розділені пробілом. Вияснити скільки разів у тексті зустрічається перше слово.
3. Задано текст, слова якого розділені %. Вияснити і надрукувати, який процент слів в тексті починається на задану букву (буква вводиться з клавіатури).
4. Розділити заданий текст на стрічки довжиною меншою, ніж 50 символів. Слова розділені пробілами. Переносити частину слова не дозволяється.
5. В заданому тексті слова розділені пробілом. Надрукувати двоскладові слова (тобто слова з двома голосними).
6. Задано текст. Визначити, з якою частотою в тексті зустрічаються букви українського алфавіту. Видати на друк частоту двох букв (назви якого вводяться з клавіатури).
7. Текст зашифрований таким чином, що всі слова записані навпаки (наприклад замість ГРУПА - АІТУРГ). Скласти програму, яка зашифровує та розшифровує текст (слова розділені пробілами).
8. В тексті, що містить не більше 50 символів, вставити рівномірно пробіли так, щоб його довжина була рівно 50 символів.
9. Задано текст, слова якого розділені пробілом. Визначити, на яку букву починається найбільше слів тексту. Вивести ці слова на друк.
10. Задано текст, слова якого розділені %. Визначити, на яку букву починається найменше слів тексту. Вивести ці слова на друк..
11. Задано текст, слова якого розділені символом #. Надрукувати в порядку спадання частоти появи букви, з яких починаються слова.
12. Задано текст, слова якого розділені пробілом. Надрукувати слово, що містить найбільшу та найменшу кількість букв (всі слова, якщо вони не єдині)
13. Задано текст, слова якого розділені пробілом. Визначити, де більше слів: в першій половині тексту чи в другій.
14. Задано текст, слова якого розділені пробілом. Визначити кількість голосних та приголосних букв у тексті.

15. Задано текст, слова якого розділені пробілом. Підрахувати кількість односкладових, двохскладових слів в тексті.
16. Задано текст, що містить різні розділові знаки (пропуск, кома, точка, знак оклику, знак запитання, дві крапки, крапка з комою). Визначити з якою частотою вони зустрічаються в тексті.
17. Задано текст, слова якого розділені пробілами. Переставити слова тексту так, щоб останнє слово стало першим, передостаннє - другим і т. д.
18. Задано текст, слова якого розділені різними розділовими знаками. Знайти в ньому всі слова, що містять найбільшу кількість голосних букв.
19. Задано текст, слова якого розділені різними розділовими знаками. Знайти в ньому всі слова, в яких доля букв "а", "б"- максимальна.
20. Задано текст. Визначити чи містить він символи , що відмінні від букв і пробіла.
21. Задано текст. Задано текст, слова якого розділені різними розділовими знаками. Знайти кількість слів, в яких співпадає перша та остання буква, і вивести їх на екран.
22. Задано текст. Знайти кількість слів, в яких дві однакові голосові букви, і вивести їх на екран.
23. Задано текст. Знищити всі символи, які не являються буквами або цифрами, і поміняти кожен велику букву малою.
24. Задано текст. Замінити всі маленькі букви одноіменними великими.
25. Задано дві стрічки. Знайти найбільшу довжину їхньої спільної підстрічки
26. Задано дві стрічки. Знайти найдовшу їхню спільну підстрічку.
27. Задано текст, що містить різні розділові знаки (пропуск, кома, крапка, знак оклику, знак запитання, двокрапка). Визначити кількість таких знаків.
28. Задано текст, слова якого розділені різними розділовими знаками. Вилучити всі знаки пунктуації.
29. Задано дві стрічки тексту. Перевірити чи збігаються за змістом ці дві стрічки, тобто мають однаковий порядок слів і знаків пунктуації, але можуть відрізнятися кількістю й розташуванням пропусків.
30. Задано текст. Вирізати з кожної групи розміщених підряд цифр, в якій більше ніж дві цифри і яка розміщена перед крапкою, всі цифри починаючи з третьої.

ЛАБОРАТОРНА РОБОТА №21: “РОБОТА ЗІ СТРІЧКАМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику використання функцій з бібліотек обробки символів та стрічок засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

робота з функціями з бібліотеки обробки символів <ctype.h>;

робота з функціями з бібліотеки обробки стрічок <string.h>;

робота з функціями зі стандартної бібліотеки введення/виведення <stdio.h>;

робота з функціями зі стандартної бібліотеки загального призначення <stdlib.h>.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_21_01.

```
/*
 *
 * Функції для виконання операцій зі стрічками
 * з бібліотеки обробки стрічок <string.h>:
 * char *strcpy(char *s1, const char *s2);
 * char *strncpy(char *s1, const char *s2, size_t n);
 * char *strcat(char *s1, const char *s2);
 * char *strncat(char *s1, const char *s2, size_t n);
 */
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    clrscr();
    char x[]="  Щасливого Вам Нового року!";
    char y[30];
    char z[15];
    printf("\n%s\n%s\n", " Демонструємо роботу функцій",
        " які копіюють стрічки:");

    // Функція char *strcpy(char *s1, const char *s2)
    // копіює стрічку s2 в масив s1. Повертає стрічку s1:
    printf("%s%s\n%s%s\n",
        "\n Стрічка в масиві x є такою:", x,
        " Стрічка в масиві y є такою:", strcpy(y,x));

    // Функція char *strncpy(char *s1, const char *s2, size_t n)
    // копіює не більше, ніж n символів зі стрічки s2 в масив s1.
    // Повертає стрічку s1:
    strncpy(z,x,14);
    z[11]='\0';
    printf(" Стрічка в масиві z є такою:%s\n",z);

    printf("\n%s\n%s\n", " Демонструємо роботу функцій",
        " які об'єднують стрічки:");
    char s1[20]="Happy ";
```

```

char s2[]="New Year";
char s3[40]="";
printf("\n  s1=%s\n  s2=%s\n", s1,s2);

// Функція char *strcat(char *s1, const char *s2)
// об'єднує стрічку s2 зі стрічкою масиву s1.
// Перший символ стрічки s2 переписує символ NULL.
// Повертає стрічку s1:
printf("  strcat(s1,s2)=%s\n", strcat(s1,s2));

// Функція char *strncat(char *s1, const char *s2, size_t n)
// об'єднує не більше, ніж n символів стрічки s2, зі стрічкою
// зі стрічкою s1.
// Перший символ стрічки s2 переписує символ NULL.
// Повертає стрічку s1:
printf("  strncat(s3,s1, 6)=%s\n", strncat(s3, s1, 6));
printf("  strcat(s3,s1)=%s\n", strcat(s3,s1));
getch();
return 0;
}

```

Результати тестування програми:

Демонструємо роботу функцій
які копіюють стрічки:

Стрічка в масиві x є такою: Щасливого Вам Нового року!
Стрічка в масиві y є такою: Щасливого Вам Нового року!
Стрічка в масиві z є такою: Щасливого

Демонструємо роботу функцій
які об'єднують стрічки:

```

s1=Happy
s2=New Year
strcat(s1,s2)=Happy New Year
strncat(s3,s1, 6)=Happy
strcat(s3,s1)=Happy Happy New Year

```

2. Ввести, відлагодити та протестувати програму під назвою lab_work_21_02.

```

/*****
/*
/* Функції порівняння стрічок з бібліотеки обробки
/* стрічок <string.h>:
/* int strcmp(const char *s1, const char *s2);
/* int strncmp(const char *s1, const char *s2, size_t n);
/*
*****/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
int main()

```

```

{clrscr();
const char *s1="Happy New Year";
const char *s2="Happy New Year";
const char *s3="Happy Holidays";

// Функція int strcmp(const char *s1, const char *s2)
// порівнює стрічку s1 зі стрічкою s2. Повертає значення:
// менше нуля - якщо s1 менше за s2;
//          нуль - якщо s1 дорівнює s2;
// більше нуля - якщо s1 більше за s2;

printf("%s%s\n%s%s\n%s%s\n\n%s%2d\n%s%2d\n%s%2d\n\n",
      "  s1= ",s1,"  s2= ",s2,"  s3= ",s3,
      "  strcmp(s1,s2)=", strcmp(s1,s2),
      "  strcmp(s1,s3)=", strcmp(s1,s3),
      "  strcmp(s3,s1)=", strcmp(s3,s1));

// Функція int strncmp(const char *s1, const char *s2)
// порівнює не більше, ніж n символів стрічки s1,
// зі стрічкою s2. Повертає значення:
// менше нуля - якщо s1 менше за s2;
//          нуль - якщо s1 дорівнює s2;
// більше нуля - якщо s1 більше за s2;
printf("%s%2d\n%s%2d\n%s%2d\n\n",
      "  strncmp(s1,s3, 6)=", strncmp(s1,s3,6),
      "  strncmp(s1,s3, 7)=", strncmp(s1,s3, 7),
      "  strncmp(s3,s1, 7)=", strncmp(s3,s1, 7));

getch();
return 0;
}

```

Результати тестування програми:

```

s1= Happy New Year
s2= Happy New Year
s3= Happy Holidays

```

```

strcmp(s1,s2)= 0
strcmp(s1,s3)= 6
strcmp(s3,s1)=-6

```

```

strncmp(s1,s3, 6)= 0
strncmp(s1,s3, 7)= 6
strncmp(s3,s1, 7)=-6

```

3. Ввести, відлагодити та протестувати програму під назвою lab_work_21_03.

```

/*****
/*
/* Функція визначення довжини стрічки з бібліотеки
/* обробки стрічок <string.h>:
/*
/*      size_t strlen(const char *s);
*/

```



```

/*                                                                 */
/*  Функція визначає довжину стрічки s. Повертає                */
/*  кількість символів, які передують завершальному нулю.      */
/*                                                                 */
/*****
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main()
{ clrscr();

    // s1 - Задана стрічка, довжину якої визначаємо:
    const char *s1="Це стрічка-тест на функцію strlen";

    // s2 - Ще одна стрічка, довжину якої визначаємо:
    const char *s2="123456789";

    // s3 - Ще одна стрічка, довжину якої визначаємо:
    const char *s3="";

    int len1, len2, len3; // довжини відповідних стрічок

    printf("\n%s\n%s\n", " Демонструємо роботу функції",
           "  size_t strlen(const char *s):");

    printf("\n%s\n%s%s\n%s%s\n",
           "  Задано такі стрічки: ",
           "  s1: ",s1,
           "  s2: ",s2,
           "  s3: ",s3);

    len1=strlen(s1);
    len2=strlen(s2);
    len3=strlen(s3);

    printf("\n%s%d\n%s%d\n%s%d\n",
           "  Довжина len1 стрічки s1 дорівнює: ",len1,
           "  Довжина len2 стрічки s2 дорівнює: ",len2,
           "  Довжина len3 стрічки s3 дорівнює: ",len3);
    getch();
    return 0;
}

```

Результати тестування програми:

Демонструємо роботу функції
size_t strlen(const char *s):

Задано такі стрічки:
s1: Це стрічка-тест на функцію strlen
s2: 123456789
s3:

Довжина len1 стрічки s1 дорівнює: 33
Довжина len2 стрічки s2 дорівнює: 9
Довжина len3 стрічки s3 дорівнює: 0

4. Ввести, відлагодити та протестувати програму під назвою lab_work_21_04.

```
/*
 *
 * Функції пошуку з бібліотеки обробки стрічок
 * <string.h>:
 * char *strchr(const char *s, int c);
 * size_t strcspn(const char *s1, const char *s2);
 * size_t strspn(const char *s1, const char *s2);
 * char *strprbk(const char *s1, const char *s2);
 * char *strrchr(const char *s, int c);
 * char *strstr(const char *s1, const char *s2);
 * char *strtok(const char *s1, const char *s2);
 */
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main()
{
    clrscr();
    // Функція char *strchr(const char *s, int c) знаходить
    // позицію першого входження символу c у стрічку s.
    // Якщо c знайдено, повертає вказівник на c в стрічці s.
    // Інакше повертається вказівник зі значенням NULL:

    const char *s1=" Це -вказівник-константа на char";
    char c1='a'; // змінна типу char
    char c2='б'; // змінна типу char
    clrscr();
    printf("\n%s\n%s\n", " Демонструємо роботу функції",
           " char *strchr(const char *s, int c):");
    if (strchr(s1, c1) != NULL)
        printf("\n%s%c%s\n%s\n",
           " Символ '\"", c1, "\"' входить в стрічку: ", s1);
    // " s1= ", s1, " s2= ", s2,
    // " Довжина початкового сегмента стрічки s1, який містить ",
    // " тільки ті символи, що не входять у стрічку s2 =",
    // strcspn(s1, s2));
    getch();

    // Функція size_t strcspn(const char *s1, const char *s2)
    // визначає і повертає довжину початкового сегмента стрічки s1,
    // який містить лише ті символи, які не входять у стрічку s2:

    const char *s3="The value is 3.14159"; // вказівник-константа на
char
    const char *s4="i1234567890"; // вказівник-константа на char
}
```

```

clrscr();
printf("\n%s\n%s\n\n", " Демонструємо роботу функції",
      " size_t strcspn(const char *s1, const char *s2):");
printf("%s%s\n%s%s\n\n%s\n%s%u",
      " s1= ", s3, " s2= ", s4,
      " Довжина початкового сегмента стрічки s1, який містить ",
      " тільки ті символи, що не входять у стрічку s2
=", strcspn(s3, s4));
getch();

// Функція size_t strspn(const char *s1, const char *s2)
// визначає і повертає довжину початкового сегмента стрічки s1,
// який містить лише ті символи, які входять у стрічку s2:

// Функція char *strprbk(const char *s1, const char *s2) знаходить
// позицію першого входження будь-якого символу стрічки s2
// у стрічці s1. Якщо символ стрічки s2 знайдено, функція
повертає
// вказівник на цей символ у стрічці s1.
// Інакше повертається вказівник зі значенням NULL:

// Функція char *strrchr(const char *s, int c ) знаходить
// позицію позицію останнього входження символу c у стрічку s.
// Якщо символ c знайдено, функція повертає
// вказівник на цей символ у стрічці s.
// Інакше повертається вказівник зі значенням NULL:

// Функція char *strstr(const char *s1, const char *s2) знаходить
// позицію першого входження стрічки s2 у стрічці s1.
// Якщо підстрічка знайдена, повертає вказівник підстрічки у
стрічці s1.
// Інакше повертається вказівник зі значенням NULL:

// Функція char *strtok(const char *s1, const char *s2) ділить
// стрічку s1 на лексеми, розділені між собою символами стрічки
s2.
// Під час першого виклику функція отримує як аргумент стрічку s1.
// У наступних викликах функції для розбиття тієї ж самої стрічки
s1
// на лексеми, як перший аргумент передається NULL.
// Під час кожного виклику функції повертається вказівник на
поточну
// лексему стрічки s1. Якщо за чергового виклику функції
з'ясовується,
// що лексем більше немає, повертається вказівник зі значенням
NULL:
getch();
return 0;
}

```

Результати тестування програми:

Демонструємо роботу функції
char *strchr(const char *s, int c):

Символ 'a' входить в стрічку:
Це -вказівник-константа на char

Демонструємо роботу функції
size_t strcspn(const char *s1, const char *s2):

s1= The value is 3.14159
s2= i1234567890

Довжина початкового сегмента стрічки s1, який містить
тільки ті символи, що не входять у стрічку s2 =10

5. Ввести, відлагодити та протестувати програму під назвою lab_work_21_05

```
/*
/*****
/* Дано стрічку s1[80]. Виділити з неї
/* підстрічку s2, яка починається з позиції n та
/* містить n символів. Для вводу різних варіантів
/* стрічки s1 скласти нескінченний цикл, вихід з
/* якого відбувається тоді, коли стрічка s1 матиме
/* вигляд "*****".
/* Розробити підпрограму-функцію для виділення
/* підстрічки заданої довжини, починаючи із
/* заданої позиції.
/*
/*****
#include <stdio.h>
#include <string.h>
#include <conio.h>
int substr(char *src, char *dest, int pos, int len);

int main()
{clrscr();
char s1[80]; // s1 - стрічка-джерело
char s2[80]; // s2 - стрічка-результат
int n,l,r; //n-позиція, l-довжина, r-параметр виведення

// Ввід і обробка стрічки s1 у нескінченному циклі::
for (;;)
{
    // Ввід стрічки s1:
    printf(" Введіть стрічку-джерело s1:\n");
    gets(s1);
    // Умова виходу з нескінченного циклу:
    if (!strcmp(s1,"*****")) break;

    // Введення позиції pos і довжини len:
    printf(" Введіть параметри pos та len:\n");
    scanf("%d%d",&n,&l);

    // Формування підстрічки за допомогою підпрограми:
    r=substr(s1,s2,n,l);
}
```

```

    // Друк результатів:
    printf("  Стрічка-джерело: s1=%s \n",s1);
    printf("  Номер позиції pos=%d та довжина len=%d\n",n,l);
    printf("  Стрічка-результат: s2=%s \n",s2);
    printf("  Параметр виведення r=%d \n\n",r);
    gets(s1);
}
getch();
return 0;
}
//-----
// Виділення підстрічки заданої довжини, починаючи із заданої
// позиції
int substr(char *src, char *dest, int pos, int len)
{
    /*
    /*
    /* Параметри:
    /* src - стрічка-джерело;
    /* dest - стрічка-результат;
    /* pos - позиція, з якої виділяється підстрічка-результат;
    /* len - задана довжина підстрічки-результату.
    /*
    /* Функція повертає значення:
    /* 0 - у випадку коректної роботи;
    /* 1 - стрічка dest має довжину, меншу за len;
    /* -1 - недопустимі значення параметрів pos або len.
    /*
    /*
    /*
    int res; // Значення, яке повертає функція
    // Встановлення вказівника src на позицію pos:
    for(; pos && *src; pos--) *src++;
    // Перевірка параметрів pos та len на коректність
    // та формування стрічки-результату у разі їх коректності:
    if (pos || (len<0)) res=-1; // параметри некоректні
    else // параметри коректні:
    {
        // Копіювання len символів зі стрічки src у стрічку dest:
        for (; len&&*src; len--) *dest++=*src++;
        // Формування значення res, яке повертає функція:
        res=len?1:0;
        // Додавання завершального нуля до стрічки-результату:
        *dest='\0';
    }
    return res;
}
}

```

6. Ввести, відлагодити та протестувати програму під назвою lab_work_21_06.

```

/*
/*
/* Дано стрічку s1[80]. Слова у цій стрічці
/* розділені одним пробілом.
/*
/* Для вводу різних варіантів стрічки s1 скласти
/* нескінченний цикл, вихід з якого відбувається
/* тоді, коли стрічка s1 матиме вигляд "*****".
/*

```

```

/*  Визначити кількість слів у заданій стрічці та */
/*  надрукувати їх.                                */
/*                                                    */
/*****/
#include <stdio.h>
#include <string.h>
#include <conio.h>
int kilstr(char *src);
int main()
{clrscr();
 char s[80];    // s - стрічка-джерело. У цій стрічці кожне слово
 char sl[80];   // має закінчуватися завершальним нулем
                // (масив стрічок)
 char *sl[20];  // sl - масив вказівників на char.
                // Він містить адреси кожного слова в стрічці sl
 int kil;       //kil-кількість слів у стрічці
 int i,j,len;   // len - довжина стрічки

// Нескінченний цикл для багаторазового тестування програми:
for (;;)
{printf("  \n Введіть задану стрічку s:\n");
 gets(s);
 if (!strcmp(s,"*****")) break;// умова виходу з циклу

// Визначення довжини len стрічки s:
 len=strlen(s);
 printf("\n  Довжина len стрічки s: %d\n",len);

// Визначення кількості пробілів у стрічці:
 kil=0;
 for (i=0; i<=len; i++)
  if (*(s+i)==' ') kil++; // поточний символ є пробілом
 kil++;                  // слів на 1 більше, ніж пробілів

// Друк кількості слів:
 printf("\n  Задана така стрічка:\n  ");
 puts(s);
 printf("\n  Кількість слів у стрічці: %d \n",kil);

 *sl=s1;
 j=0;
 printf("\n  Слова заданої стрічки:\n  ");
 for (i=0; i<=len; i++)
  if (*(s+i)!=' ')
  {*(sl+i)= *(s+i);
   printf("%c",*(sl+i));
  }
  else
  {*(sl+i)='\0';
   printf("\n  ");
   *(sl+j)=s1+i+1;
   j++;
  }
}

```

```

// Розв'язок задачі за допомогою масиву вказівників
printf("\n Задана така стрічка:\n ");
for (i=0; i<=len; i++) printf("%c", *(s1+i));
printf("\n");
printf("\n Слова заданої стрічки:\n ");
for (j=0; j<kil; j++)
{for (i=0; *(*s1+j)+i!='\0';i++)
    printf("%c", *(*s1+j)+i));
    printf("\n ");
}
}
getch();
return 0;
}

```

Результати тестування програми:

Введіть стрічку-джерело s1:
 Луцьк - обласний центр
 Введіть параметри pos та len:
 8 8
 Стрічка-джерело: s1=Луцьк - обласний центр
 Номер позиції pos=8 та довжина len=8
 Стрічка-результат: s2=обласний
 Параметр r=0

Введіть стрічку-джерело s1:

```

7. Ввести, відлагодити та протестувати програму під назвою lab_work_21_07.
/*****
/*
/* Ввести з клавіатури прізвище та ім'я (наприклад,
/* Петренко Іван).
/* Утворити і надрукувати стрічку "Привіт, Іван
/* Петренко!".
/*
*****/
#include <stdio.h>
#include <string.h>
#include <conio.h>
int main()
{clrscr();
char s[80], a[80]="Привіт, ";
int i,len,p;
printf(" Введіть прізвище та ім'я: ");
gets(s);

clrscr();
printf(" Введені прізвище та ім'я:\n ");
puts(s);
len=strlen(s); // Знаходимо довжину стрічки s
printf("\n Довжина len стрічки s:%d\n",len);

```

```
// Шукаємо позицію p першого пробілу в стрічці s:
p=0;
while ((s[p]!=' ') && (p<len)) p++;

if (p>len)
{printf("\n Прізвище та ім'я введені неправильно!\n");
getc();
return 1;
}
else
{strcat(a, s+p+1);      // До стрічки a додаємо ім'я
strcat(a, " ");        // До стрічки a додаємо пробіл
len=strlen(a);         // Знаходимо довжину стрічки a
strncpy(a+len,s,p);    // До стрічки a додаємо прізвище
strcpy(a+len+p, "!");  // До стрічки a додаємо знак оклику
printf("\n Результуюча стрічка:\n ");
puts(a);               // Друкуємо стрічку a
}
getch();
return 0;
}
```

Результати тестування програми:

Введені прізвище та ім'я:
Кузава Олег
Довжина заданої стрічки s дорівнює: 11

Результуюча стрічка:
Привіт, Олег Кузава

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

1. Дано рядок символів. Перетворити його, видаливши кожен символ * і повторивши кожен символ, відмінний від *.
2. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Підрахувати кількість слів в даному рядку.
3. Дано рядок символів в якому є двокрапка. Одержати всі символи, розташовані до першої двокрапки включно.
4. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Знайти кількість слів, у яких перша і остання букви співпадають.
5. Даний рядок символів в якому є двокрапка. Одержати всі символи, розташовані після першої двокрапки.
6. Даний рядок символів. Підрахувати найбільшу кількість розташованих підряд пропусків.
7. Даний рядок символів. З'ясувати, чи вірно, що в рядку є п'ять підряд букв e.
8. Даний рядок символів. Визначити число входжень в рядок групи букв abc.
9. Даний рядок символів. Визначити число входжень в рядок групи букв мама.
10. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Підрахувати кількість букв а в останньому слові.

11. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Знайти кількість слів, що починаються з букви б.
12. Даний рядок символів. Визначити число входжень в рядок групи букв aba.
13. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Знайти яке-небудь слово, що починається з букви а.
14. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Знайти довжину найкоротшого слова.
15. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Знайти довжину найдовшого слова.
16. Даний рядок символів в якому є двокрапка. Одержати всі символи, розташовані між першою і другою двокрапкою. Якщо другої двокрапки немає, то одержати всі символи, розташовані після єдиної наявної двокрапки.
17. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Слова що читаються зліва на право і справа на ліво однаково називатимемо паліндромами. Знайти довжину найбільшого паліндрома.
18. Даний рядок символів в якому є двокрапка. Одержати всі символи, розташовані між першою і другою двокрапкою. Якщо другої двокрапки немає, то одержати всі символи, розташовані до єдиної наявної двокрапки.
19. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Слова що читаються зліва на право і справа на ліво однаково називатимемо паліндромами. З'ясувати, чи вірно, що кожне слово, що не є паліндромом, має парну довжину.
20. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Слова що читаються зліва на право і справа на ліво однаково називатимемо паліндромами. Знайти найкоротший паліндром.
21. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Слова що читаються зліва на право і справа на ліво однаково називатимемо паліндромами. Знайти кількість паліндромов в рядку.
22. Даний рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Слова що читаються зліва на право і справа на ліво однаково називатимемо паліндромами. Знайти слова-паліндроми що мають парну довжину.
23. Дано рядок символів. Вилучіть з нього пропуски та коми, решту символів продублюйте.
24. Дано рядок символів. Замініть в ньому всі літери о на а та виведіть його в зворотному порядку.
25. Дано рядок символів. Групи символів, розділені пропусками (одним або декількома), називатимемо словами. Змініть текст, залишивши лише по одному пропуску між словами.
26. Заданий текст, що містить різні розділові знаки (пропуск, кома, крапка, знак оклику, знак запитання, двокрапка). Визначити кількість таких знаків.
27. Задано текст, слова якого розділені пробілом. Підрахувати кількість односкладових, двоскладових, трьохскладових і т.д. слів у тексті.
28. Заданий текст, що містить різні розділові знаки(пропуск, кома, крапка, знак оклику, знак запитання, двокрапка). Визначити з якою частотою вони зустрічаються у тексті.
29. Задано текст, слова якого розділені пробілом. Переставити слова у тексті так, щоб останнє слово стало першим, передостаннє - другим і т.д.
- 30.** Задано текст, слова якого розділені пробілом. Вивести на екран найдовше слово.

ЛАБОРАТОРНА РОБОТА №22: “РОБОТА ЗІ СТРУКТУРАМИ ТА ОБ’ЄДНАННЯМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику роботи зі структурами та об’єднаннями засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

оголошення структур у програмах мовою C/C++;
формування та сортування масиву структур;
оголошення об’єднань у програмах мовою C/C++;
використання об’єднань для вирішення конкретних завдань.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_22_01.

```
/*
*****
/*
/*      Формування і сортування масиву структур.
/*      Завдання. Сформувати масив з kst структур з такими
/*      полями:прізвище, ім'я; група; номер студентського квитка;
/*      середній бал. Впорядкувати цей масив за спаданням значень
/*      середнього балу. Вивести на друк впорядкований масив.
/*      Завдання реалізувати засобами мови C.
/*
/*
*****
#include <stdio.h>
#include <conio.h>
#define KST 25 //

int main(void)
{clrscr();

    struct student
    {char name[35];
      char grup[10];
      long number;
      double rating;
    }starr[KST], stud, *pst, *pst1, *pst2;

    int k, kst;
    double inrating;
    printf("\n Кількість студентів:");
    scanf("%d",&kst);
    getchar();
    puts("\t Дані:");
    for (k=0; k<kst; k++)
    {printf("\n%2d. Прізвище,  ім'\': ", k+1);
      gets(starr[k].name);
      printf("      : ");
      gets((starr+k)->grup);
      printf("      Номер студентського квитка: ");
      scanf("%ld",&starr[k].number);
```

```

    printf("    Середній бал: ");
    scanf("%lf",&inrating);
    getchar();
    (starr+k)->rating=inrating;
}

for (k=0; k<kst; k++)
{pst1=starr;
pst2=starr+1;
for ( ; pst2<starr+kst-k; pst1++, pst2++)
    if (pst1->rating<pst2->rating)
        {stud=*pst1;
         *pst1=*pst2;
         *pst2=stud;
        }
}

puts("\n\t Список студентів:\n");
for (k=0, pst=starr; k<kst; k++,pst++)
    printf("%2d. %-30.27s%-8s%8ld%8.2lf\n",k+1,pst->name,
           pst->grup, pst->number, pst->rating);

getch();
return 0;
}

```

Результати тестування програми:

Кількість студентів:2

Дані:

1. Прізвище, ім'я: Петренко Антон
Група: KCM11
Номер студентського квитка: 12345
Середній бал: 4.56
2. Прізвище, ім'я: Черкасенко Анжела
Група: KCM11
Номер студентського квитка: 13689
Середній бал: 4.74

Список студентів:

- | | | | |
|----------------------|-------|-------|------|
| 1. Черкасенко Анжела | KCM11 | 13689 | 4.74 |
| 2. Петренко Антон | KCM11 | 12345 | 4.56 |

2. Ввести, відлагодити та протестувати програму під назвою lab_work_22_02.

```

/*****
/*
/* Формування масиву структур.
/* Структура містить такі поля: марка, рік випуску машини,
/* ціна машини. Вивести на екран інформацію про всі машини і
/* додатково про ті, ціна яких менша, ніж 3000$.
/*

```

```

/*****/
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <iomanip.h>
#define n 3 //

struct Avtoprice
{char marka[15];
 int year, price;
};

int Auto_desh(struct Avtoprice *p);

void main()
{clrscr();
 int k;

 Avtoprice a1[n];
 for (int i=0; i<n; i++)
 {cout<<endl<<" Уведіть марку машини - ";
  cin>>a1[i].marka;
  cout<<" Уведіть рік випуску машини - ";
  cin>>a1[i].year;
  cout<<" Уведіть ціну машини в $ - ";
  cin>>a1[i].price;
 }

 cout<<"\n"<<" Фірма пропонує такі машини:"<<"\n";
 cout<<"Назва Рік Ціна "<<"\n";
 cout<<"машини випуску машини "<<"\n";
 for (i=0; i<n; i++)

 cout<<setprecision(10)<<a1[i].marka<<"\t"<<a1[i].year<<"\t"<<a1[i].p
rice<<"\n";

 k=Auto_desh(a1);
 if (k==0)
  cout<<"endl<<" Таких автомобілів немає"
  getch();

 }

//-----
int Auto_desh(struct Avtoprice *p)
{int k;
 cout<<"Машини дешевші, ніж 3000$: \n";
 cout<<"Назва Рік Ціна "<<"\n";
 cout<<"машини випуску машини "<<"\n";
 k=0;
 for (int i=0; i<n; i++)
  if (p[i].price < 3000)

```

```

        {cout<<setprecision(10)<<p[i].marka<<"\t"<<p[i].year<<"\t"
"<<p[i].price<<"\n";
        k++;
    }
    return k;
}

```

Результати тестування програми:

Уведіть дані про 3 машини:

```

Уведіть марку машини          - BMV
Уведіть рік випуску машини    - 2000
Уведіть ціну машини в $      - 20000

Уведіть марку машини          - Mersedes
Уведіть рік випуску машини    - 2005
Уведіть ціну машини в $      - 10000

Уведіть марку машини          - Shcoda
Уведіть рік випуску машини    - 1995
Уведіть ціну машини в $      - 5000

```

Фірма пропонує такі машини:

Назва машини	Рік випуску	Ціна машини
BMV	2000	20000
Mersedes	2005	10000
Shcoda	1995	5000

Машини дешевші, ніж 3000\$:

Назва машини	Рік випуску	Ціна машини
-----------------	----------------	----------------

Таких автомобілів немає

3. Ввести, відлагодити та протестувати програму під назвою lab_work_22_03.

```

/*****
/*
/*      Об'єднання.
/*      . Завдання. Створити об'єднання number_parts з такими
/*      полями:
/*      long number - довге ціле число;
/*      lip - структура типу LIP;
/*      unsigned char bytes[4] - масив з 4-х елементів.
/*      Структура LIP має такі поля:
/*      unsigned int low - молодша частина довгого числа;
/*      unsigned int high - старша частина довгого числа;
/*      Використовуючи об'єднання, виділити обидві частини
*/

```

```

/* довгого числа і розкласти його на байти. */
/* */
/*****/
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

//Оголошення нового типу даних - структури longint_parts
typedef struct longint_parts
{unsigned int low, high;
}LIP; // LIP (тип даних) - структура типу longint_parts

//Оголошення об'єднання number_parts
union number_parts
{long number;
LIP lip;
unsigned char bytes[4];
} un={0x1a2b3c4d1}; //un - об'єкт-об'єднання типу number_parts

int main(void)
{clrscr();

int i;
i=sizeof(struct longint_parts);
printf("\n Розмір пам'яті під шаблон структури longint_parts:
%d байтів",i);
i=sizeof(LIP);
printf("\n Розмір пам'яті під змінну-структуру LIP:
%d байтів",i);
i=sizeof(union number_parts);
printf("\n Розмір пам'яті під шаблон об'єднання number_parts:
%d байтів",i);
i=sizeof(un);
printf("\n Розмір пам'яті під змінну-об'єднання un:
%d байтів \n",i);

printf("\n Довге ціле - %#0.10lx",un.number);
printf("\n Молодше слово - %#0.6x",un.lip.low);
printf("\n Старше слово - %#0.6x",un.lip.high);
puts("\n\n Байти:");
printf("\n");
for (i=0; i<4; i++)
printf("%#08.4x ", (unsigned)un.bytes[i]);
getch();
return 0;
}

```

Результати тестування програми:

```

Розмір пам'яті під шаблон структури longint_parts: 4 байтів
Розмір пам'яті під змінну-структуру LIP: 4 байтів
Розмір пам'яті під шаблон об'єднання number_parts: 4 байтів

```

Розмір пам'яті під змінну-об'єднання un: 4 байтів
Довге ціле - 0x1a2b3c4d
Молодше слово - 0x3c4d
Старше слово - 0x1a2b
Байти:
0x4d 0x3c 0x2b 0x1a

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Створити, відлагодити та протестувати програму для обробки заданого згідно з індивідуальним варіантом масиву структур, що містить задані у варіанті поля.

Варіант 1

1. № п/п
2. Номер поїзда
3. Товарний чи пасажирський
4. Тепловоз чи електровоз
5. Куди відправляється (Київ, Львів, Одеса)
6. Кількість вагонів

Визначити номери пасажирських поїздів, які відправляються в Одесу

Варіант 2

1. № п/п
2. Вид н/д роботи (бюджетна, договірна)
3. Науковий керівник (прізвище)
4. Закінчена чи не закінчена
5. Об'єм фінансування в тис.
6. Рік впровадження в практику

Визначити і надрукувати керівників закінчених договірних робіт

Варіант 3

1. № п/п
2. Прізвище студента
3. Назва групи (ОА-11, ЕП-11, МС-11, АГ-11)
4. Рік закінчення школи
5. Стаж роботи (Так, ні)
6. Заборгованість з дисципліни (хімія, фізика, ТКМ)

Визначити прізвища студентів групи МС 11, які мають заборгованість з двох предметів

Варіант 4

1. № п /п
2. Прізвище
3. Рік народження
4. Вид спорту (футбол, хокей, шахи, теніс)
5. Країна (Україна, Литва, Чехія, Китай)
6. Зайняте місце (1-8)

Визначити футболістів України старших за 28 років

Варіант 5

1. № п /п
2. Назва предмету (фізика, хімія, ТММ)
3. Залік чи екзамен

4. Прізвище викладача
 5. Група (ОА-11, СМ-11, ЕП-11)
 6. Середній бал групи (2, 3, 4, 5)
- Визначити групу, що має найвищий середній бал з фізики

Варіант 6

1. № п/п
 2. Тип оцінювання (залік, екзамен)
 3. Предмет (математика, фізика, графіка)
 4. Форма здачі (усна, письмова)
 5. Оцінка (2, 3, 4, 5)
 6. Форма навчання (заочна, стаціонарна)
- Визначити предмет з найкращого письмовою оцінкою на стаціонарі

Варіант 7

1. №.п/п
 2. Назва виробу (холодильник, телевізор)
 3. Рік випуску
 4. Вітчизняний чи імпортований
 5. Гарантія в роках (один, два, п'ять)
 6. Форма оплати (повністю, частково)
- Визначити холодильники, в яких закінчилась гарантія

Варіант 8

1. № п/п
 2. Назва книги
 3. Рік видання
 4. Мова (українська, російська, англійська)
 5. Прізвище автора
 6. Форма користування (власна, бібліотека)
- Визначити книги, видані українською мовою, які є в бібліотеці

Варіант 9

1. № п/п
 2. Назва фірми (Зоря, МФК, TOR.LTD)
 3. Профіль (виробництво, посередництво, торгівля)
 4. Рік заснування
 5. Пільги (так, ні)
 6. Прізвище керівника
- Визначити фірми та їх керівників, зайнятих виробництвом

Варіант 10

1. № п/п
 2. Марка авто (Лада, Волга, Ауді)
 3. Дата випуску
 4. Країна випуску (Росія, ФРН)
 5. Прізвище власника
 6. Техогляд (так, ні)
- Визначити марки авто, випущених в ФРН, які не пройшли техогляд

Варіант 11

1. № п/п
 2. Характеристика поля (чорнозем, нечорнозем)
 3. Площа поля в га
 4. Тип добрив (азотні, калійні)
 5. Потреба обробки (так, ні)
 6. Власник чи орендатор
- Визначити власників чорноземних полів, яким потрібні азотні добрива

Варіант 12

1. № п/п
 2. Прізвище
 3. Рік народження
 4. Вид спорту (лижі, шахи, теніс)
 5. Розряд (перший, другий, майстер)
 6. Яку країну представляє (Україна, Росія, Італія)
- Визначити спортсменів України з тенісу, старших за 28 років

Варіант 13

1. № п/п
 2. Назва закладу (інститут, університет, технікум)
 3. Тип (технічний, гуманітарний)
 4. Вступні екзамени (2 або 3)
 5. Адреса (місто)
 6. План набору (в цифрах)
- Визначити університети технічного напрямку, що мають 3 екзамени

Варіант 14

1. № п/п
 2. Назва авто (Ауді, Нива, БМВ, Опель)
 3. Дизельний чи карбюраторний
 4. Рік випуску
 5. Пробіг (в тис. км.) (S - Прізвище власника)
- Визначити власників дизельних авто, випущених до 1993 р.

Варіант 15

1. № п/п
 2. Прізвище
 3. Звання (рядовий, майор, генерал)
 4. Рід військ (піхота, авіація)
 5. Рік народження
 6. Нагороди (має, немає)
- Визначити рядових піхоти, які мають нагороду

Варіант 16

1. № п/п
 2. Назва континенту
 3. Назва країни
 4. Столиця
 5. Населення (млн)
 6. Частина кулі (північна, південна)
- Визначити столиці північної півкулі з населенням більше 1 млн.

Варіант 17

1. № п/п
2. Назва санаторію (Нива, Черемош, Лаванда)
3. Номер (одномісний, двомісний, тримісний)
4. Місто (Моршин, Трускавець)
5. Корпус (1-5)
6. Прізвище пацієнта

Визначити наявність одномісних номерів в Черемоші.

Варіант 18

1. № п/п
2. Прізвище
3. Спеціальність (вчитель, лікар, інженер)
4. Місце роботи (школа, лікарня, завод)
5. Рік працевлаштування
6. Відпустка (планова, позапланова)

Визначити позапланових відпустників - інженерів

Варіант 19

1. № п/п
2. Вид продукції (хліб, овочі, молоко)
3. Одиниця вимірювання (кг, л, штуки)
4. Спосіб реалізації (в роздріб, оптом)
5. Термін реалізації (день, тиждень, місяць)
6. Термін закінчення договору (1998 - 2005 р)

Визначити кількість молочних товарів в л., що продаються за 1 день.

Варіант 20

1. № п/п
2. Прізвище
3. Факультет (ТФ, БФ, ЕФ, МБФ)
4. Група
5. Курс
6. Форма навчання (стаціонар, заочна)
7. Визначити студентів групи, що належать до МБФ.

Варіант 21

1. № п/п
2. Прізвище
3. Рік народження
4. Стать
5. Характер роботи (постійна, тимчасова)

Визначити працівників, старших за 55, які працюють тимчасово.

Варіант 22

1. № п/п
2. Тип квартири (окрема, комунальна)
3. Зручності (повні, часткові)
4. Номер ЖЕКу (1 - 10)
5. Прізвище власника
6. Адреса (вулиця)

Визначити власників квартир ЖЕКу № 5.

Варіант 23

1. № п/п
2. Прізвище
3. Посада (асистент, доцент, професор)
4. Факультет (ЕФ, ІКФ, ТФ, МБФ)
5. Предмет (фізика, механіка, хімія)
6. Семестр

Визначити прізвище професорів МБФ.

Варіант 24

1. № п/п
2. Назва журналу (Дзвін, Дніпро, Київ, Вітчизна)
3. Рік видання
4. Редактор (Коротич, Драч, Дзюба)
5. Реалізація (підписка, вроздріб)
6. Місто видання (Київ, Львів, Луцьк)

Визначити редакторів і назви журналів, які видаються у Львові.

Варіант 25

1. № п/п
2. Прізвище
3. Стать (чол., жін ,)
4. Соціальний статус (робітник, селянин, інтелігенція)
5. Користується транспортом (трамвай, тролейбус, авто)
6. Чи використана відпустка (так, ні)

Визначити прізвище чол., які користуються авто.

Варіант 26

1. № п/п
2. Прізвище
3. Ім'я
4. Рік народження
5. Факультет (МБФ, ІКФ, ЕФ, ЗФ)
6. Форма навчання (очна, заочна, вечірня)

Визначити студентів -заочників МБФ.

Варіант 27

1. № п/п
2. Назва книги
3. Рік видання
4. Видавництво (Вежа, Стир, Дніпро)
5. Автор
6. Ціна (в гривнях)

Визначити назву книг видавництва Дніпро, виданих до 1988 р.

Варіант 28

1. № п/п
2. Прізвище футболіста
3. Функція (воротар, захисник, півзахисник, нападаючий)
4. Назва команди (Волинь, Верес, Нива, Зоря)
5. Рік народження

6. Ліга (вища, перша)
Визначити воротарів команд вищої ліги.

Варіант 29

1. № п/п
2. Тип магазину (гастроном, універмаг, аптека)
3. Державний чи приватний
4. Прізвище власника
5. Загальна площа
- 5 Адреса (вулиця)

Визначити адресу приватних аптек.

Варіант 30

1. № п/п
2. Прізвище
3. Ім'я
4. Група (ОТ- 11, ФОФ- 11, ТМ- 11)
5. Місце проживання (райцентр, село, обласний центр)
6. Рік народження

Визначити студентів групи ТМ-11, які проживають в селі

ЛАБОРАТОРНА РОБОТА №23: “РОБОТА З ФАЙЛАМИ ПОСЛІДОВНОГО ТА ПРЯМОГО ДОСТУПУ ЗАСОБАМИ МОВИ C”

МЕТА РОБОТИ

1. Опанувати методику роботи з файлами послідовного та прямого доступу засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика роботи з файлами послідовного доступу засобами мови C/C++;

методика роботи з файлами прямого доступу засобами мови C/C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_23_01.

```
/*
 *
 * Робота з файлом послідовного доступу.
 * Створити файл послідовного доступу clients.dat,
 * передбачивши три режими роботи з ним:
 * створення і запису даних, зчитування та друку
 * даних, дозапису даних у кінець файла.
 * Файл має містити записи з такими полями:
 * Номер рахунку, Прізвище власника, Борг.
 */
#include <stdio.h>
#include <iostream.h>
#include <conio.h>

void main()
{ clrscr();
  int rah;          // номер рахунку
  char priz[15];    // прізвище власника рахунку
  float borg;       // баланс (борг)
  int ozn;          // режим роботи з файлом

  FILE *fPtr;       // вказівник на файл clients.dat

  // Вибір режиму роботи з файлом:
  printf("\n%s\n%s\n%s",
    " Введіть 0, якщо хочете створити і записати дані у файл!",
    " Введіть 1, якщо хочете зчитати дані з файлу!",
    " Введіть 2, якщо хочете дозаписати дані у файл: ");
  scanf("%d",&ozn);

  if (ozn==0)
  { // Режим створення файлу і запису даних:
    // Відкриття файлу для створення і запису даних:
    fPtr=fopen("clients.dat","w");
    if (fPtr==NULL)
      printf("\n Файл не можливо відкрити!\n");
    else
    {
```

```

printf("\n Введіть номер рахунку, прізвище власника,
        баланс!\n");
printf("\n Якщо всі дані введені, виконайте команду
        Ctrl+Z!\n");
printf(" ? ");
scanf("%d%s%f", &rah, &priz, &borg);

// Поки не введені всі дані у файл,, виконуйте:
while ( !feof(stdin))
{
    // Збереження поточного запису у файлі:
    fprintf(fPtr, "%d %s %.2f\n", rah, priz, borg);
    printf(" ? ");
    scanf("%d%s%f", &rah, &priz, &borg);
}
fclose(fPtr); // Закриття файла
getch();
}
else if (ozn==1)
{
    // Відкриття файла для читання даних:
    fPtr=fopen("clients.dat", "r");
    if (fPtr==NULL)
        printf("\n Файл не можливо відкрити!\n");
    else
    {
        printf("\n Номер рахунку    Прізвище власника    Борг\n");
        fscanf(fPtr, "%d%s%f", &rah, &priz, &borg);

        // Поки не надруковані всі дані з файлу, виконуйте:
        while ( !feof(fPtr))
        {
            printf("      %-10d      %-12s      %8.2f\n", rah, priz, borg);

            // Друк поточного запису у файлі:
            fscanf(fPtr, "%d%s%f", &rah, &priz, &borg);
        }
        fclose(fPtr); // Закриття файла
        getch();
    }
}
else if (ozn==2)
{
    // Відкриття файла для дозапису даних:
    fPtr=fopen("clients.dat", "a");
    if (fPtr==NULL)
        printf("\n Файл не можливо відкрити!\n");
    else
    {
        printf("\n Введіть номер рахунку, прізвище власника,
                баланс!\n");
        printf("\n Якщо всі дані введені, виконайте команду
                Ctrl+Z!\n");
        printf(" ? ");
    }
}

```

```

scanf("%d%s%f", &rah, &priz, &borg);

// Поки не введені всі дані у файл, виконуйте:
while ( !feof(stdin))
{
    // Збереження поточного запису у файлі:
    fprintf(fPtr, "%d %s %.2f\n", rah, priz, borg);
    printf(" ? ");
    scanf("%d%s%f", &rah, &priz, &borg);
}
fclose(fPtr); // Закриття файла
getch();
}
}
}

```

Результати тестування програми:

Введіть 0, якщо хочете створити і записати дані у файл!
Введіть 1, якщо хочете зчитати дані з файлу!
Введіть 2, якщо хочете дозаписати дані у файл: 0

Введіть номер рахунку, прізвище власника, баланс!
Якщо всі дані введені, виконайте команду Ctrl+Z!
? 1 Петренко 56.54
? 2 Власюк 77.21
? 3 Славський 89.65
? ^Z

Введіть 0, якщо хочете створити і записати дані у файл!
Введіть 1, якщо хочете зчитати дані з файлу!
Введіть 2, якщо хочете дозаписати дані у файл: 1

Номер рахунку	Прізвище власника	Борг
1	Петренко	56.54
2	Власюк	77.21
3	Славський	89.65

Введіть 0, якщо хочете створити і записати дані у файл!
Введіть 1, якщо хочете зчитати дані з файлу!
Введіть 2, якщо хочете дозаписати дані у файл: 2

Введіть номер рахунку, прізвище власника, баланс!
Якщо всі дані введені, виконайте команду Ctrl+Z!
? 4 Бринський 44.25
? ^Z

Введіть 0, якщо хочете створити і записати дані у файл!
Введіть 1, якщо хочете зчитати дані з файлу!
Введіть 2, якщо хочете дозаписати дані у файл: 1

Номер рахунку	Прізвище власника	Борг
1	Петренко	56.54
2	Власюк	77.21
3	Славський	89.65
4	Бринський	44.25

2. Ввести, відлагодити та протестувати програму під назвою lab_work_23_02.

```

/*****/
/*
/* Робота з файлом прямого доступу.
/* Створити файл прямого доступу credit.dat,
/* передбачивши три режими роботи з ним:
/* створення і запису даних, зчитування та друку
/* даних, дозапису даних у кінець файла.
/* Файл має містити записи з такими полями:
/* Номер рахунку, Прізвище власника, Борг.
/*
/*****/
#include <stdio.h>
#include <iostream.h>
#include <conio.h>

struct ClientData
{
    int rah;          // номер рахунку
    char priz[15];    // прізвище власника рахунку
    char im[10];      // ім'я власника рахунку
    float borg;       // баланс (борг)
};

void main()
{
    clrscr();

    int ozn;          // режим роботи з файлом

    struct ClientData client={0,"", "",0.00};

    FILE *fPtr;       // вказівник на файл clients.dat

    // Вибір режиму роботи з файлом:
    printf("\n%s\n%s\n%s",
        " Введіть 0, якщо хочете створити і записати дані у файл!",
        " Введіть 1, якщо хочете зчитати дані з файлу!",
        " Введіть 2, якщо хочете дозаписати дані у файл:  ");
    scanf("%d",&ozn);

    if (ozn==0)
    {
        // Режим створення і запису даних
        // Відкриття файла для створення і запису

```



```

fPtr=fopen("credit.dat","wb");

if (fPtr==NULL)
printf("\n Файл не можливо відкрити!\n");
else
{
printf("\n%s\n%s\n%s\n%s\n ",
      " Введіть номер рахунку -",
      " Якщо всі дані введені,",
      " введіть нуль!");
scanf("%d",&client.rah);

// Поки не введені усі дані у файл, виконуйте:
while (client.rah!=0)
{
printf("\n Введіть прізвище власника, ім'я власника,
      баланс!");
printf("\n Якщо всі дані введені, введіть нуль!\n");
printf(" ? ");

// Ввід даних з клавіатури у поля робочого запису client:
fscanf(stdin,"%s%s%f",&client.priz, &client.im,
      &client.borg);

// Пошук позиції поточного запису у файлі:
fseek(fPtr, (client.rah-1)*sizeof(struct ClientData),
      SEEK_SET);

// Збереження поточного запису у файлі:
fwrite(&client, sizeof(struct ClientData), 1, fPtr);

// Ввід номера наступного рахунку:
printf("\n%s\n%s\n%s\n%s\n ",
      " Введіть номер рахунку -",
      " (число між 1 та 1000)!",
      " Якщо всі дані введені,",
      " введіть нуль!");
scanf("%d",&client.rah);

}
fclose(fPtr);
getch();
}
else if (ozn==1)
{
// Режим зчитування та друку даних

// Відкриття файла для зчитування:
fPtr=fopen("credit.dat","rb");

if (fPtr==NULL)
printf("\n Файл не можливо відкрити!\n");

```

```

else
{
    // Друк шапки файла:
    printf("\n Номер      Прізвище      Ім'я      Борг");
    printf("\n рахунку      клієнта      клієнта      клієнта\n");

    // Поки не досягнуто кінця файла,
    // на який вказує fPtr, виконуйте:
    while ( !feof(fPtr))
    {
        if (client.rah!=0) // Якщо не досягнуто кінця файла:
            // Друк поточного запису
            printf("      %-6d      %-6s      %-6s      %6.2f\n",
                client.rah,client.priz,client.im,client.borg);

            // Зчитування даних з файла у запис client:
            fread(&client, sizeof(struct ClientData),1,fPtr);
        }

        fclose(fPtr); // Закриття файла
        getch();
    }
}
else if (ozn==2)
{
    // Режим дозапису даних у кінець файла
    // Відкриття файла для дозапису даних:
    fPtr=fopen("credit.dat","ab");

    if (fPtr==NULL)
        printf("\n  Файл не можливо відкрити!\n");
    else
    {
        printf("\n%s\n%s\n%s\n%s\n  ",
            "  Введіть номер рахунку -",
            "  (число між 1 та 1000)!",
            "  Якщо всі дані введені,",
            "  введіть нуль!");
        scanf("%d",&client.rah);

        // Поки не введені усі дані у файл, виконуйте:
        while ( client.rah!=0)
        {
            printf("\n  Введіть Прізвище власника, ім'я власника,
                баланс!");
            printf("\n  Якщо всі дані введені, введіть нуль!\n");
            printf("  ?  ");

            // Ввід даних з клавіатури у поля робочого запису
            client:
            fscanf(stdin,"%s%s%f",&client.priz, &client.im,
                &client.borg);

```

```

// Пошук позиції поточного запису у файлі:
fseek(fPtr, (client.rah-1)*sizeof(struct ClientData),
      SEEK_SET);

// Збереження поточного запису у файлі:
fwrite(&client, sizeof(struct ClientData),1,fPtr);

// Ввід номера наступного рахунку:
printf("\n%s\n%s\n%s\n%s\n  ",
      " Введіть номер рахунку -",
      " (число між 1 та 1000)!",
      " Якщо всі дані введені,",
      " введіть нуль!");
scanf("%d",&client.rah);

    }
    fclose(fPtr);
    getch();
}
}
}

```

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Створити, відлагодити та протестувати файл послідовного доступу засобами мови C для обробки заданого у лабораторній роботі №22 за індивідуальним варіантом масиву структур.

Завдання 2. Створити, відлагодити та протестувати файл прямого доступу засобами мови C++ для обробки заданого у лабораторній роботі №22 за індивідуальним варіантом масиву структур.

ЛАБОРАТОРНА РОБОТА №24: “КОНСТРУЮВАННЯ КЛАСІВ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику конструювання класів засобами мови C++.
2. Опанувати методику роботи з об'єктами засобами мови C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика конструювання класів засобами мови C++;

методика роботи з об'єктами засобами мови C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_24_01.

```
/*
*****
/*
Класи та об'єкти в C++.
/*
Завдання. Створити клас Complex для роботи з
/*
комплексними числами, у якому передбачити такі
/*
елементи-функції:
/*
InputCompl()
/*
- введення дійсної та уявної частини комплексного числа;
/*
OutputCompl()
/*
- виведення дійсної та уявної частини
/*
і модуля комплексного числа;
/*
ModCompl()
/*
- обчислення модуля комплексного числа;
/*
SumCompl()
/*
- додавання двох комплексних чисел;
/*
DifCompl()
/*
- віднімання двох комплексних чисел;
/*
MultCompl()
/*
- множення двох комплексних чисел;
/*
DivCompl()
/*
- ділення двох комплексних чисел.
/*
*****
#include <iostream.h>
#include <conio.h>
#include <math.h>
class Complex
{public:
void InputCompl(float *Re, float *Im);
void OutputCompl(float Re, float Im, float Mod);
void ModCompl(float Re, float Im, float *Mod);
void SumCompl(float Re1, float Im1, float Re2, float Im2,
float *Re, float *Im, float *Mod);
void DifCompl(float Re1, float Im1, float Re2, float Im2,
float *Re, float *Im, float *Mod);
void MultCompl(float Re1, float Im1, float Re2, float Im2,
float *Re, float *Im, float *Mod);
void DivCompl(float Re1, float Im1, float Re2, float Im2,
float *Re, float *Im, float *Mod);
```

```

};
// Функція введення дійсної та уявної частини комплексного числа
void Complex::InputCompl(float *Re, float *Im)
{cout<<" Введіть дійсну частину комплексного числа: ";
  cin>>*Re;
  cout<<" Введіть уявну частину комплексного числа: ";
  cin>>*Im;
}

// Функція виведення дійсної та уявної частини і модуля
// комплексного числа
void Complex::OutputCompl(float Re, float Im, float Mod)
{cout<<"=====\\n";
  cout<<" Дійсна частина комплексного числа "<<Re<<"\\n";
  cout<<" Уявна частина комплексного числа: "<<Im<<"\\n";
  cout<<" Модуль комплексного числа: "<<Mod<<"\\n";
  cout<<"=====\\n";
}

// Функція обчислення модуля комплексного числа
void Complex::ModCompl(float Re, float Im, float *Mod)
{
  *Mod=pow((Re*Re + Im*Im), 1./2.);
}

// Функція додавання двох комплексних чисел
void Complex::SumCompl(float Re1, float Im1, float Re2, float Im2,
                      float *Re, float *Im, float *Mod)
{*Re=Re1+Re2;
  *Im=Im1+Im2;
  *Mod=pow((*Re)*(*Re)+(*Im)*(*Im), 1./2.);
}

// Функція віднімання двох комплексних чисел
void Complex::DifCompl(float Re1, float Im1, float Re2, float Im2,
                      float *Re, float *Im, float *Mod)
{*Re=Re1-Re2;
  *Im=Im1-Im2;
  *Mod=pow((*Re)*(*Re)+(*Im)*(*Im), 1./2.);
}

// Функція множення двох комплексних чисел
void Complex::MultCompl(float Re1, float Im1, float Re2,
                      float Im2, float *Re, float *Im, float *Mod)
{*Re=Re1*Re2-Im1*Im2;
  *Im=Re1*Im2+Re2*Im1;
  *Mod=pow((*Re)*(*Re)+(*Im)*(*Im), 1./2.);
}

// Функція ділення двох комплексних чисел
void Complex::DivCompl(float Re1, float Im1, float Re2, float Im2,
                      float *Re, float *Im, float *Mod)
{*Re=(Re1*Re2+Im1*Im2)/(Re2*Re2+Im2*Im2);

```

```

*Im=(-Re1*Im2+Re2*Im1)/(Re2*Re2+Im2*Im2);
*Mod=pow ((*Re)*(*Re)+(*Im)*(*Im), 1./2.);
}

void main()
{clrscr();
Complex Compl;
float R1,R2,I1,I2,Rez,Imz,M1,M2,M;

// Введення першого комплексного числа
cout<<"\n Введення першого комплексного числа: \n";
Compl.InputCompl (&R1,&I1);

// Введення другого комплексного числа
cout<<"\n Введення другого комплексного числа: \n";
Compl.InputCompl (&R2,&I2);
clrscr();

// Обчислення модуля першого комплексного числа
Compl.ModCompl (R1,I1,&M1);

// Обчислення модуля другого комплексного числа
Compl.ModCompl (R2,I2,&M2);

// Виведення першого комплексного числа
cout<<"\n Перше комплексне число: \n";
Compl.OutputCompl (R1,I1,M1);

// Виведення другого комплексного числа
cout<<"\n Друге комплексне число: \n";
Compl.OutputCompl (R2,I2,M2);
getch();
clrscr();

// Сума двох комплексних чисел
Compl.SumCompl (R1,I1,R2,I2, &Rez, &Imz, &M);
cout<<"\n Сума двох комплексних чисел: \n";
Compl.OutputCompl (Rez,Imz,M);
getch();
clrscr();

// Різниця двох комплексних чисел
Compl.DifCompl (R1,I1,R2,I2, &Rez, &Imz, &M);
cout<<"\n Різниця двох комплексних чисел: \n";
Compl.OutputCompl (Rez,Imz,M);
getch();
clrscr();

// Добуток двох комплексних чисел
Compl.MultCompl (R1,I1,R2,I2, &Rez, &Imz, &M);
cout<<"\n Добуток двох комплексних чисел: \n";
Compl.OutputCompl (Rez,Imz,M);
getch();
}

```

```
clrscr();

// Частка двох комплексних чисел
Compl.DivCompl(R1,I1,R2,I2, &Rez, &Imz, &M);
cout<<"\n Частка двох комплексних чисел: \n";
Compl.OutputCompl(Rez,Imz,M);
getch();
}
```

Результати тестування програми:

Введення першого комплексного числа:

Введіть дійсну частину комплексного числа: 1

Введіть уявну частину комплексного числа: 2

Введення другого комплексного числа:

Введіть дійсну частину комплексного числа: 3

Введіть уявну частину комплексного числа: 4

Перше комплексне число:

```
=====
Дійсна частина комплексного числа      1
Уявна частина комплексного числа:      2
Модуль комплексного числа:              2.236068
=====
```

Друге комплексне число:

```
=====
Дійсна частина комплексного числа      3
Уявна частина комплексного числа:      4
Модуль комплексного числа:              5
=====
```

Сума двох комплексних чисел:

```
=====
Дійсна частина комплексного числа      4
Уявна частина комплексного числа:      6
Модуль комплексного числа:              7.211102
=====
```

Різниця двох комплексних чисел:

```
=====
Дійсна частина комплексного числа      -2
Уявна частина комплексного числа:      -2
Модуль комплексного числа:              2.828427
=====
```

Різниця двох комплексних чисел:

```
=====
Дійсна частина комплексного числа      -2
Уявна частина комплексного числа:      -2
Модуль комплексного числа:              2.828427
=====
```

```
=====
Добуток двох комплексних чисел:
=====
```

```
Дійсна частина комплексного числа    -5
Уявна частина комплексного числа:    10
Модуль комплексного числа:           11.18034
=====
```

```
=====
Частка двох комплексних чисел:
=====
```

```
Дійсна частина комплексного числа    0.44
Уявна частина комплексного числа:    0.08
Модуль комплексного числа:           0.447214
=====
```

2. Ввести, відлагодити та протестувати програму під назвою lab_work_24_02.

```
/*
 *
 *  Конструювання класів та об'єктів засобами мови C++.
 *  Завдання. Сформувати клас FootTeams з такими
 *  елементами:
 *  закриті елементи класу:
 *  char title[64] - назва команди;
 *  char best_player[64] - прізвище кращого гравця;
 *  char country[64] - країна;
 *  int rating - рейтинг команди;
 *  відкриті елементи класу:
 *  FootTeams(char *, char *, int, char *) -
 *  - конструктор з чотирма параметрами;
 *  FootTeams() - конструктор без параметрів;
 *  void show() -
 *  функція виведення на друк даних про команду.
 *  Передбачити різні форми звертання до об'єктів класу.
 */
#include <iostream.h>
#include <string.h>
#include <conio.h>

int a;

// Визначення класу FootTeams
class FootTeams
{public:
    // Конструктор з чотирма параметрами
    FootTeams(char *, char *, int, char *);
    // Конструктор без параметрів
    FootTeams();
    // Функція виведення на друк даних про команду
    void show();

private:
    char title[64];
}
```



```

    char best_player[64];
    char country[64];
    int rating;
}; //кінець оголошення класу FootTeams

void FootTeams::show()
{cout<<"\n Назва команди:"<<title<<endl;
  cout<<" Країна:" <<country<<endl;
  cout<<" Рейтинг команди:" <<rating<<endl;
  cout<<" Кращий гравець:" <<best_player<<endl;
}

FootTeams::FootTeams(char *t, char *c, int r, char *b)
{strcpy(title,t);
  strcpy(country,c);
  rating=r;
  strcpy(best_player,b);
}

FootTeams::FootTeams()
{rating=118;
  strcpy(title, "Зоря Луганськ");
  strcpy(country, " Україна");
  strcpy(best_player," Тьоркін");
}

void main()
{clrscr();

  FootTeams d1("Барселона", " Іспанія", 1, " Мессі");
  FootTeams d2("Динамо Київ", " Україна",32," Блохін");
  FootTeams d3;

  d1.show();
  d2.show();
  d3.show();

  getch();
}

```

Результати тестування програми:

Назва команди:Барселона
 Країна: Іспанія
 Рейтинг команди:1
 Кращий гравець: Мессі

Назва команди:Динамо Київ
 Країна: Україна
 Рейтинг команди:32
 Кращий гравець: Блохін

Назва команди:Зоря Луганськ

Країна: Україна
Рейтинг команди:118
Кращий гравець: Тьоркін

3. Ввести, відлагодити та протестувати програму під назвою lab_work_24_03.

```
/*
*****
*/
/*
Конструювання класів та об'єктів засобами мови C++.
*/
/*
Завдання. Сформувати клас LNTU з такими компонентами:
*/
/*
закриті компоненти класу
*/
/*
char title[64] - назва факультету;
*/
/*
int kil - Загальна кількість студентів ;
*/
/*
int vyb - Кількість вибувших студентів;
*/
/*
void copyt(char *t) - функція копіювання стрічки;
*/
/*
відкриті елементи класу:
*/
/*
LNTU(char *, int, int) - конструктор з 3-ма парам.;
*/
/*
LNTU(char *, int) - конструктор з 2-ма парам.;
*/
/*
LNTU() - конструктор без параметрів;
*/
/*
int get_kil() - повернення загальної кількості студ.;
*/
/*
int get_vyb() - повернення кількості вибувших студ.;
*/
/*
void get_title(char *t) - повернення назви факульт.;
*/
/*
void set_title(char *t) - збереження назви факульт.;
*/
/*
void set_kil(int k) - збереження заг. кільк. студ.;
*/
/*
void set_vyb(int kv) - збереження кільк. виб. студ.;
*/
/*
void print() - друк результатів.
*/
/*
Передбачити різні форми звертання до об'єктів класу.
*/
/*
*****
*/
#include <iostream.h>
#include <string.h>
#include <conio.h>

// Визначення класу LNTU
class LNTU
{
//Закриті елементи класу LNTU:
private:
char title[64]; // Назва факультету
int kil; // Загальна кількість студентів
int vyb; // Кількість вибувших студентів
void copyt(char *t); // Функція копіювання стрічки

//Відкриті елементи класу LNTU:
public:
LNTU(char *, int, int); //Конструктор з трьома параметрами
LNTU(char *, int); //Конструктор з двома параметрами
LNTU(); //Конструктор без параметрів
int get_kil();
int get_vyb();
void get_title(char *t);
void set_kil(int k);
void set_title(char *t);
void set_vyb(int kv);
void print();
}
```

```

}; // Кінець оголошення класу LNTU

//Опис закритих елементів-функцій класу LNTU:

// Функція копіювання стрічки t в title (назву факультету)
void LNTU::copyt(char *t) {strcpy(title, t);}

//Опис відкритих елементів-функцій класу LNTU:

// Конструктор класу LNTU з трьома параметрами -
// ініціалізує назву факультету,
// загальну кількість студентів та кількість вибувших студентів
LNTU::LNTU(char *t, int k, int kv)
{
    copyt(t);
    kil=k;
    vyb=kv;
}

// Конструктор класу LNTU з двома параметрами -
// ініціалізує назву факультету
// та загальну кількість студентів
LNTU::LNTU(char *t, int k)
{
    copyt(t);
    kil=k;
};

// Конструктор класу LNTU без параметрів -
// ініціалізує назву факультету (очищує),
// загальну кількість студентів (обнулює),
// та кількість вибувших студентів (обнулює)
LNTU::LNTU()
{
    copyt("");
    kil=0;
    vyb=0;
};

// Функція, яка забезпечує виведення на друк назви факультету,
// загальної кількості студентів та кількості вибувших студентів
void LNTU::print()
{
    cout<<"\n Назва факультету: "<<title<<endl;
    cout<<" Загальна кількість студентів: "<<kil<<endl;
    cout<<" Кількість вибувших студентів: "<<vyb<<endl;
};

// Функція, яка повертає назву факультету
void LNTU::get_title(char *t) {strcpy(t,title)};

// Функція, яка повертає загальну кількість студентів
int LNTU::get_kil() {return kil};

// Функція, яка повертає кількість вибувших студентів
int LNTU::get_vyb() {return vyb};

```

```

// Функція, яка зберігає назву факультету
void LNTU::set_title(char *t) {copyt(t);};

// Функція, яка зберігає загальну кількість студентів
void LNTU::set_kil(int k) {kil=k;};

// Функція, яка зберігає кількість вибувших студентів
void LNTU::set_vyb(int kv) {vyb=kv;};

void main()
{clrscr();
 int k, kv, a[3], m;
 char s[70];

 cout<<"\n  Передача даних через конструктор з трьома
        параметрами.";
 cout<<"\n  Всі три параметри - константи:\n ";
 LNTU d1("ФКНІТ",10,2);
 d1.print();
 cout<<"\n  Натисніть яку-небудь клавішу, щоб продовжити... ";
 getch();

 clrscr();

 cout<<"\n  Передача даних через конструктор з трьома
        параметрами.";
 cout<<"\n  Перший параметр - константа,";
 cout<<"\n  а два інші параметри мають бути введені
        з клавіатури:\n ";
 cout<<"\n  Факультет ФБ: ";
 cout<<"\n  Введіть загальну кількість студентів: ";
 cin>>k;
 cout<<"  Введіть кількість вибувших студентів: ";
 cin>>kv;
 LNTU d2("ФБ",k,kv);
 d2.print();
 cout<<"\n  Натисніть яку-небудь клавішу, щоб продовжити... ";
 getch();
 clrscr();

 cout<<"\n  Передача даних через конструктор з трьома
        параметрами.";
 cout<<"\n  Перший параметр - константа,";
 cout<<"\n  а два інші параметри мають бути введені
        з клавіатури:\n ";
 cout<<"\n  Факультет ФОФ: ";
 cout<<"\n  Введіть загальну кількість студентів: ";
 cin>>k;
 cout<<"  Введіть кількість вибувших студентів: ";
 cin>>kv;
 LNTU d3("ФОФ",k,kv);

```

```

d3.print();
cout<<"\n  Натисніть яку-небудь клавішу, щоб продовжити... ";
getch();
clrscr();

cout<<"\n  Передача даних через конструктор з двома
    параметрами.";
cout<<"\n  Перший параметр - константа,";
cout<<"\n  а другий параметр має бути введений
    з клавіатури:\n ";
cout<<"\n  Факультет МБФ: ";
cout<<"\n  Введіть загальну кількість студентів: ";
cin>>k;
LNTU d4("МБФ",k);
d4.print();
cout<<"\n  Кількість вибувших студентів - неправильна!:";
cout<<"\n  У функцію d4.print передано два параметри,
    а треба три!\n";
cout<<"\n  Додатково введіть кількість вибувших студентів: ";
cin>>kv;
d4.set_vyb(kv);
d4.print();
cout<<"\n  Тепер всі дані правильні!\n";
cout<<"\n  Натисніть яку-небудь клавішу, щоб продовжити... ";
getch();
clrscr();

cout<<"\n  Передача даних через конструктор без параметрів.";
cout<<"\n  Всі аргументи функції d5.print() мають бути
    введені поза конструктором:\n ";
cout<<"\n  Факультет ТФ: ";
LNTU d5;
cout<<"\n  Введіть назву факультету: ";
cin>>s;
cout<<"  Введіть загальну кількість студентів: ";
cin>>k;
cout<<"  Введіть кількість вибувших студентів: ";
cin>>kv;
d5.set_title(s);
d5.set_kil(k);
d5.set_vyb(kv);
d5.print();
cout<<"\n  Натисніть яку-небудь клавішу, щоб продовжити... ";
getch();
clrscr();

cout<<"\n  Передача даних через конструктор з трьома
    параметрами.";
cout<<"\n  Перший параметр - константа,";
cout<<"\n  а два інші параметри мають бути введені
    з клавіатури:\n ";
cout<<"\n  Факультет ФЕПЕС: ";
cout<<"\n  Введіть загальну кількість студентів: ";

```

```

cin>>k;
cout<<" Введіть кількість вибувчих студентів: ";
cin>>kv;
LNTU d6("ФЕПЕС",k,kv);
d6.print();
cout<<"\n Загальна кількість студентів: "<<d6.get_kil()<<endl;
cout<<"\n (Визначено за допомогою функції d6.get_kil()) \n";
cout<<"\n Натисніть яку-небудь клавішу, щоб продовжити... ";
getch();
clrscr();

// Створення динамічного масиву
// Доступ до об'єктів здійснюється за допомогою вказівника
cout<<"\n Обробка даних по трьох факультетах : ";
LNTU *d7;
d7=new LNTU[3];
m=0;
for (int i=0; i<3; i++)
{cout<<"\n Введіть назву "<<i+1<<" -го факультету: ";
cin>>s;
cout<<" Введіть кількість студентів: ";
cin>>k;
cout<<" Введіть кількість вибувчих студентів: ";
cin>>kv;
d7[i].set_title(s);
d7[i].set_kil(k);
d7[i].set_vyb(kv);
}
for (i=0;i<3; i++)
{d7[i].print();
m=m+d7[i].get_vyb();
}
cout<<"\n Всього вибувчих студентів: "<<m<<endl;
cout<<"\n Натисніть яку-небудь клавішу, щоб закінчити роботу...
";
getch();
}

```

Результати тестування програми:

Передача даних через конструктор з трьома параметрами.
Всі три параметри – константи:

Назва факультету: ФКНІТ
Загальна кількість студентів: 989
Кількість вибувчих студентів: 2

Натисніть яку-небудь клавішу, щоб продовжити...

Передача даних через конструктор з трьома параметрами.
Перший параметр – константа,
а два інші параметри мають бути введені з клавіатури:

Факультет ФБ:

Введіть загальну кількість студентів: 545

Введіть кількість вибувших студентів: 3

Назва факультету: ФБ

Загальна кількість студентів: 545

Кількість вибувших студентів: 3

Натисніть яку-небудь клавішу, щоб продовжити...

Передача даних через конструктор з трьома параметрами.

Перший параметр - константа,

а два інші параметри мають бути введені з клавіатури:

Факультет ФОФ:

Введіть загальну кількість студентів: 654

Введіть кількість вибувших студентів: 3

Назва факультету: ФОФ

Загальна кількість студентів: 654

Кількість вибувших студентів: 3

Натисніть яку-небудь клавішу, щоб продовжити...

Передача даних через конструктор з двома параметрами.

Перший параметр - константа,

а другий параметр має бути введений з клавіатури:

Факультет МБФ:

Введіть загальну кількість студентів: 678

Назва факультету: МБФ

Загальна кількість студентів: 678

Кількість вибувших студентів: 3

Кількість вибувших студентів - неправильна!:

У функцію d4.print передано два параметри, а треба три!

Додатково введіть кількість вибувших студентів: 5

Назва факультету: МБФ

Загальна кількість студентів: 678

Кількість вибувших студентів: 5

Тепер всі дані правильні!

Натисніть яку-небудь клавішу, щоб продовжити...

Передача даних через конструктор без параметрів.

Всі аргументи функції d5.print() мають бути поза конструктором:

Факультет ТФ:

Введіть назву факультету: ТФ

Введіть загальну кількість студентів: 567

Введіть кількість вибувших студентів: 6

Назва факультету: ТФ

Загальна кількість студентів: 567

Кількість вибувших студентів: 6

Натисніть яку-небудь клавішу, щоб продовжити...

Передача даних через конструктор з трьома параметрами.

Перший параметр - константа,

а два інші параметри мають бути введені з клавіатури:

Факультет ФЕПЕС:

Введіть загальну кількість студентів: 679

Введіть кількість вибувших студентів: 3

Назва факультету: ФЕПЕС

Загальна кількість студентів: 679

Кількість вибувших студентів: 3

Загальна кількість студентів: 679

(Визначено за допомогою функції d6.get_kil())

Натисніть яку-небудь клавішу, щоб продовжити...

Обробка даних по трьох факультетах :

Введіть назву 1 -го факультету: ФБ

Введіть кількість студентів: 567

Введіть кількість вибувших студентів: 2

Введіть назву 2 -го факультету: ФОФ

Введіть кількість студентів: 789

Введіть кількість вибувших студентів: 2

Введіть назву 3 -го факультету: ФКНІТ

Введіть кількість студентів: 897

Введіть кількість вибувших студентів: 1

Назва факультету: ФБ

Загальна кількість студентів: 567

Кількість вибувших студентів: 2

Назва факультету: ФОФ

Загальна кількість студентів: 789

Кількість вибувших студентів: 2

Назва факультету: ФКНІТ

Загальна кількість студентів: 897

Кількість вибувших студентів: 1

Натисніть яку-небудь клавішу, щоб закінчити роботу...

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Створити клас з полями та методами згідно Вашого варіанту. Оголосити кілька об'єктів цього класу, використовуючи конструктори з параметрами, без параметрів та конструктор копіювання. Продемонструвати роботу всіх методів класу.

<u>1.СТУДЕНТ</u> поля: ім'я – char* курс – int стать - int(bool) методи: зміна курсу введення імені введення статі виведення інформації про об'єкт	<u>2. СЛУЖБОВЕЦЬ</u> поля: ім'я – char* вік – int стаж – int методи: введення імені введення віку введення стажу виведення інформації про об'єкт
<u>3. ІСПИТ</u> поля: прізвище студента – char* номер білету – int оцінка – int методи: введення прізвища студента введення номера білету введення оцінки виведення інформації про об'єкт	<u>4. АДРЕСА</u> поля: прізвище – char* вулиця – char* номер будинку – int методи: введення прізвища введення вулиці та номера будинку зміну адреси виведення інформації про об'єкт
<u>5. ТОВАР</u> поля: назва – char* кількість – int вартість – float методи: введення назви товару зміну кількості товару зміну вартості товару виведення інформації про об'єкт	<u>6. ВИРІБ</u> поля: назва – char* шифр – char* кількість – int методи: введення назви введення шифру зміну кількості виведення інформації про об'єкт
<u>7 БІБЛІОТЕКА</u> поля: назва книги – char* автор – char* вартість – float методи: введення назви книги введення автора книги зміну ціни книги виведення інформації про об'єкт	<u>8.КАДРИ</u> поля: ім'я працівника – char* номер цеху-int розряд- int методи: введення імені зміну цеху зміну розряду виведення інформації про об'єкт

<u>9. КВИТАНЦІЯ</u> поля: номер – int дата – int сума – float методи: введення номеру квитанції введення числа видачі квитанції введення суми до сплати виведення інформації про об'єкт	<u>10. АВТОМОБІЛЬ</u> поля: марка – char* потужність – int ціна – float методи: введення марки введення потужності та ціни виведення інформації про об'єкт
<u>11. КРАЇНА</u> поля: назва – char* форма правління – char* площа – float методи: введення країни зміна форми правління введення площі країни виведення інформації про об'єкт	<u>12. ЦЕХ</u> поля: назва – char* начальник – char* кількість працівників – int методи: введення назви цеху зміна начальника цеху зміна кількості працівників виведення інформації про об'єкт
<u>13. КОРАБЕЛЬ</u> поля: ім'я – char* водотоннажність – int прізвище власника - char* методи: введення назви корабля введення водотоннажності зміна власника виведення інформації про об'єкт	<u>14. ПЕРСОНА</u> поля: ім'я – char* вік – int стать – int(bool) методи: введення імені введення віку введення статі виведення інформації про об'єкт
<u>15. ТВАРИНА</u> поля: назва – char* клас – char* вага – int методи: введення назви введення класу зміна ваги виведення інформації про об'єкт	

ЛАБОРАТОРНА РОБОТА №25: “РОБОТА З КЛАСАМИ З РІЗНИМИ КОНСТРУКТОРАМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику конструювання класів засобами мови C++.
2. Опанувати методику роботи з об'єктами засобами мови C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика конструювання класів засобами мови C++;

методика роботи з об'єктами засобами мови C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_25_01.

```
/*
*****
*/
/*
Конструювання класів та об'єктів.
*/
/*
Завдання.
*/
/*
Сконструювати структуру client з такими полями:
*/
/*
int number - номер рахунк клієнта;
*/
/*
STRING name[11] - прізвище клієнта;
*/
/*
float balance - баланс клієнта.
*/
/*
Сконструювати клас Client з такими полями:
*/
/*
Закриті елементи класу Client:
*/
/*
client m[N]- масив структур типу client;
*/
/*
void swop(int,int) - функція обміну значень;
*/
/*
Відкриті елементи класу Client:
*/
/*
Client() - конструктор класу;
*/
/*
~Client()- деструктор класу;
*/
/*
void enter() - функція введення даних;
*/
/*
void print() - функція виведення даних;
*/
/*
void sort() - функція сортування даних;
*/
/*
void sort1() - функція сортування даних
з ручним вибором параметра
сортування.
*/
/*
Доступ до елементів класу Client здійснити
за допомогою об'єкта ob.
*/
*****
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <iomanip.h>
#include <process.h>

#define N 3 // Кількість банківських клієнтів
#define ESC 0x1b

typedef char STRING; // Новий тип даних STRING

//Оголошення структури client
```

```

struct client
{int number;           // Номер рахунку клієнта
  STRING name[11];     // Прізвище клієнта
  float balance;       // Баланс клієнта
};

//Оголошення базового класу Client
class Client
{private:              // Закриті елементи класу за замовчуванням
  client m[N];         // Масив структур типу client
  void swop(int,int);  // Обмін значенням

public:                // Відкриті елементи класу Client
  Client();            // Конструктор класу Client
  ~Client();           // Деструктор класу Client
  void enter();        // Функція введення даних
  void print();        // Функція виведення даних
  void sort(int);      // Сортування масиву
  void sort1();        // Сортування масиву з ручним введенням
                      //індекса
                      // до 10 спроб і виведенням на екран
}; // Кінець оголошення базового класу Client

// Опис конструкторів та деструкторів класу Client
Client::Client() {}
Client::~~Client() {cout<<endl;} // звільнення буфера пам'яті

// Опис інших функцій класу Client

// Функція обміну значень комірок m[i] та m[j] масиву;
void Client::swop(int i, int j)
{client c=m[i];
  m[i]=m[j];
  m[j]=c;
}

// Функція введення значень елементів масиву структур
void Client::enter()
{for (int i=0; i<N; i++)
  {cout<<endl;
   cout<<" Введіть номер рахунку клієнта:      ";
   cin>>m[i].number;
   cout<<" Введіть прізвище клієнта:          ";
   cin>>m[i].name;
   cout<<" Введіть баланс клієнта:            ";
   cin>>m[i].balance;
  };
}

// Функція виведення на друк значень елементів масиву структур
void Client::print()
{for (int i=0; i<N; i++)
  {cout<<"\n Структура N"<<setw(3)<<i+1

```

```

        <<"\n number ="<<setw(4)<<m[i].number
        <<"\n name   ="<<setw(11)<<m[i].name
        <<"\n balance="<<setw(10)<<setprecision(3)
        <<m[i].balance<<endl;
    }
}

// Функція сортування за ознакою ind значень елементів масиву
// структур
void Client::sort(int ind)
{for (int i=0; i<N-1; i++)
    {for (int j=i+1; j<N; j++)
        {switch(ind)
            {case 1:  if (m[i].number>m[j].number) swop(i,j);break;
              case 2:  if (strcmp(m[i].name,m[j].name) >0)
                        swop(i,j);break;
              case 3:  if (m[i].balance>m[j].balance)
                        swop(i,j);break;
              default: cout<<"\n Введено помилковий параметр"<<endl;
                        goto mm;
            };
        };
    };
    mm: ;
}

// Функція сортування значень елементів масиву структур
// з ручним введенням ознаки сортування ind
void Client::sort1()
{clrscr();
    for (int i=0; i<10; i++)
    {cout<<"\n  Виберіть параметр сортування: \n";
      cout<<"\n   1      - за номером рахунку";
      cout<<"\n   2      - за ім'ям";
      cout<<"\n   3      - за величиною балансу";
      cout<<"\n   5      - Вихід  \n";

      int ind;
      cin>>ind;

      if ( ind == 5) exit(1);

      cout<<"\n Результат сортування за параметром "<<ind<<" : \n";
      sort(ind);
      print();
    };
}

int main()
{clrscr();

    Client ob;  //Об'єкт класу Client

```

```

ob.enter(); //Введення значень елементів масиву структур
clrscr();

ob.print(); // Виведення значень елементів масиву структур
getch();
clrscr();

cout<<"\n Результат роботи функції sort(2): \n";
cout<<"\n яка сортує масив структур за параметром 2
      (за ім'ям): \n";
ob.sort(2); // Сортування значень елементів масиву структур за
           // ознакою 2
ob.print(); // Виведення значень елементів відсортованого
           // масиву структур
getch();
clrscr();

cout<<"\n Результат роботи функції sort1()";
cout<<"\n Вибір режиму сортування здійснюється вручну";
cout<<"\n Для продовження натисніть яку-небудь клавішу! \n";
getch();
ob.sort1(); //Виведення значень елементів відсортованого
           //масиву структур
           //з ручним вибором режиму сортування
getch();
return 0;
}

```

Результати тестування програми:

Введіть номер рахунку клієнта:	12345
Введіть прізвище клієнта:	Петренко
Введіть баланс клієнта:	34.65
Введіть номер рахунку клієнта:	23456
Введіть прізвище клієнта:	Тарасенко
Введіть баланс клієнта:	65.65
Введіть номер рахунку клієнта:	13579
Введіть прізвище клієнта:	Катренко
Введіть баланс клієнта:	45.54

```

Запис N 1
number =12345
name    =   Петренко
balance=    34.65

```

```

Запис N 2
number =23456
name    =   Тарасенко
balance=    65.65

```

```

Запис N 3

```

```
number =13579
name    =    Катренко
balance=      45.56
```

Результат роботи функції sort(2):
яка сортує масив структур за параметром 2 (за ім'ям):

```
Запис N 1
number =13579
name    =    Катренко
balance=      45.56
```

```
Запис N 2
number =12345
name    =    Петренко
balance=      34.65
```

```
Запис N 3
number =23456
name    =    Тарасенко
balance=      65.65
```

Результат роботи функції sort1()
Вибір режиму сортування здійснюється вручну
Для продовження натисніть яку-небудь клавішу!

Виберіть параметр сортування:

- 1 - за номером рахунку
 - 2 - за ім'ям
 - 3 - за величиною балансу
 - 5 - Вихід - ESC
- 1

Результат сортування за параметром 1

```
Запис N 1
number =12345
name    =    Петренко
balance=      34.65
```

```
Запис N 2
number =13579
name    =    Катренко
balance=      45.56
```

```
Запис N 3
number =23456
name    =    Тарасенко
balance=      65.65
```

2. Ввести, відлагодити та протестувати програму під назвою lab_work_25_02.
/*****/

```

/*                                                    */
/*  Конструювання класів та об'єктів засобами мови C++. */
/*  Завдання. Сформувати клас Money з такими компонентами: */
/*  закриті елементи класу - немає; */
/*  відкриті елементи класу: */
/*  Money(int a,int b,int c) - конструктор з 3-ма парам.; */
/*  Money() - конструктор без параметрів; */
/*  int dollar - кількість доларів на поточному рахунку; */
/*  int grivna - кількість гривень на поточному рахунку; */
/*  int rubl - кількість рублів на поточному рахунку; */
/*  void PrintMoney() - */
/*      функція друку залишку на рахунку; */
/*  void CurrExchange() - */
/*      функція перерахунку валют за курсом обміну. */
/*  Передбачити різні форми звертання до об'єктів класу. */
/*                                                    */
/*****/
#include <iostream.h>
#include <conio.h>

// Оголошення класу Money
class Money
{public:
    int dollar, grivna, rubl;

    // Друк залишку на рахунку (долари, гривні, рублі):
    void PrintMoney();

    // Функція перерахунку валют за курсом обміну
    void CurrExchange();

    // Конструктор1 (без параметрів)
    Money() {}

    // Конструктор2 (з трьома параметрами)
    Money(int a, int b, int c)
    {dollar=a;
     grivna=b;
     rubl=c;
    };
}; // Кінець оголошення класу Money

// Оголошення функцій класу Money

// Функція виведення на друк поточного рахунку
void Money::PrintMoney()
{cout<<"\n  Ваш поточний рахунок: \n";
 cout<<"\n  Долар:  "<<dollar;
 cout<<"\n  Гривня: "<<grivna;
 cout<<"\n  Рубль:  "<<rubl<<"\n";
}

// Функція перерахунку валют за курсом обміну

```



```

void Money::CurrExchange()
{int z, k, kdg, kdr, kgd, kgr, krd, krg;
do
{m:cout<<"\n  Визначіться, будь ласка, з операцією!";
cout<<"\n  1. Міняти долари на гривні ";
cout<<"\n  2. Міняти долари на рублі ";
cout<<"\n  3. Міняти гривні на долари ";
cout<<"\n  4. Міняти гривні на рублі ";
cout<<"\n  5. Міняти рублі на долари ";
cout<<"\n  6. Міняти рублі на гривні \n ";
cin>>z;
if ((z==1) || (z==2) || (z==3) || (z==4) || (z==5) || (z==6))
{cout<<"\n  Яку суму хочете обміняти? ";
cin>>k;
}
else
{cout<<"\n  Введено неправильний код операції.";
cout<<"\n  Для продовження натисніть яку-небудь
клавiшу...";

getch();
clrscr();
PrintMoney();
goto m;
}

// Якщо міняти долари на гривні
if (z==1)
{if (dollar>=k)
dollar=dollar-k;
else
{cout<<"\n На рахунку немає такої суми! \n";
goto m;
};
cout<<"\n  Введіть курс обміну долар-гривня: ";
cin>>kdg;
grivna=grivna+k*kdg;
};

// Якщо міняти долари на рублі
if (z==2)
{if (dollar>=k)
dollar=dollar-k;
else
{cout<<"\n На рахунку немає такої суми! \n";
goto m;
};
cout<<"\n  Введіть курс обміну долар-рубль: ";
cin>>kdr;
rubl=rubl+k*kdr;
};

// Якщо міняти гривні на долари

```

```

if (z==3)
{if (grivna>=k)
    grivna=grivna-k;
    else
        {cout<<"\n На рахунку немає такої суми! \n";
            goto m;
        };
    cout<<"\n Введіть курс обміну гривня-долар: ";
    cin>>kgd;
    dollar=dollar+k*kgd;
};

// Якщо міняти гривні на рублі
if (z==4)
{if (grivna>=k)
    grivna=grivna-k;
    else
        {cout<<"\n На рахунку немає такої суми! \n";
            goto m;
        };
    cout<<"\n Введіть курс обміну гривня-рубль: ";
    cin>>kgr;
    rubl=rubl+k*kgr;
};

// Якщо міняти рублі на долари
if (z==5)
{if (rubl>=k)
    rubl=rubl-k;
    else
        {cout<<"\n На рахунку немає такої суми! \n";
            goto m;
        };
    cout<<"\n Введіть курс обміну рубль-долар: ";
    cin>>krd;
    dollar=dollar+k*krd;
};

// Якщо міняти рублі на гривні
if (z==6)
{if (rubl>=k)
    rubl=rubl-k;
    else
        {cout<<"\n На рахунку немає такої суми! \n";
            goto m;
        };
    cout<<"\n Введіть курс обміну рубль-гривня: ";
    cin>>kgk;
    grivna=grivna+k*kgk;
};

PrintMoney();

```

```

        cout<<"\n  Визначіться, будь ласка, з наступною операцією!";
        cout<<"\n  0. Обмін завершуємо?";
        cout<<"\n  1. Обмін продовжуємо? \n  ";
        cin>>z;
        if (z==0)
            clrscr();
    } while (z!=0);
};

main ()
{clrscr();

    cout<<"\n  Демонстрація роботи Конструктора 1:";

    Money *p1=new Money(100, 250, 300);

    p1->PrintMoney();
    p1->CurrExchange();
    p1->PrintMoney();
    getch();
    clrscr();

    cout<<"\n  Демонстрація роботи Конструктора 2:";

    Money *p2=new Money();

    cout<<"\n  Долар:";
    cin>>p2->dollar;
    cout<<"  Гривня:";
    cin>>p2->grivna;
    cout<<"  Рубль:";
    cin>>p2->rubl;

    p2->CurrExchange();
    p2->PrintMoney();

    cout<<"\n  Роботу програми завершено.";
    getch();
    return 0;
}

```

Результати тестування програми:

Демонстрація роботи Конструктора1:
Ваш поточний рахунок:

Долар: 100
Гривня: 250
Рубль: 300

Визначіться, будь ласка, з операцією!
1. Міняти долари на гривні
2. Міняти долари на рублі

3. Міняти гривні на долари
 4. Міняти гривні на рублі
 5. Міняти рублі на долари
 6. Міняти рублі на гривні
- 8

Введено неправильний код операції.
Для продовження натисніть яку-небудь клавішу...

Ваш поточний рахунок:

Долар: 100
Гривня: 250
Рубль: 300

Визначіться, будь ласка, з операцією!

1. Міняти долари на гривні
 2. Міняти долари на рублі
 3. Міняти гривні на долари
 4. Міняти гривні на рублі
 5. Міняти рублі на долари
 6. Міняти рублі на гривні
- 1

Яку суму хочете обміняти? 120

На рахунку немає такої суми!

Визначіться, будь ласка, з операцією!

1. Міняти долари на гривні
 2. Міняти долари на рублі
 3. Міняти гривні на долари
 4. Міняти гривні на рублі
 5. Міняти рублі на долари
 6. Міняти рублі на гривні
- 1

Яку суму хочете обміняти? 20

Введіть курс обміну долар-гривня: 8

Ваш поточний рахунок:

Долар: 80
Гривня: 410
Рубль: 300

Визначіться, будь ласка, з наступною операцією!

0. Обмін завершуємо?
1. Обмін продовжуємо?

0

Ваш поточний рахунок:

Долар: 80
Гривня: 410
Рубль: 300

Демонстрація роботи Конструктора2:
Долар:100
Гривня:200
Рубль:300

Визначіться, будь ласка, з операцією!

1. Міняти долари на гривні
 2. Міняти долари на рублі
 3. Міняти гривні на долари
 4. Міняти гривні на рублі
 5. Міняти рублі на долари
 6. Міняти рублі на гривні
- 1

Яку суму хочете обміняти? 20

Введіть курс обміну долар-гривня: 8

Ваш поточний рахунок:

Долар: 80
Гривня: 410
Рубль: 300

Визначіться, будь ласка, з наступною операцією!

0. Обмін завершуємо?
1. Обмін продовжуємо?

0

Роботу програми завершено.

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Створити, відлагодити та протестувати програму, у якій розробити клас для обробки масиву структур, створеного у лабораторній роботі №22. Передбачити, що цей клас виконуватиме функції введення та виведення масиву структур, а також функції їх сортування.

ЛАБОРАТОРНА РОБОТА №26: “КОНСТРУЮВАННЯ БАЗОВИХ ТА ПОХІДНИХ КЛАСІВ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику конструювання базових та похідних класів засобами мови C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика конструювання базових та похідних класів засобами мови C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_26_01.

```
/*
*****
*/
/*
Базові та похідні класи. Наслідування.
*/
/*
Завдання.
*/
/*
Сконструювати структуру client з полями:
*/
/*
int number - Номер рахунку клієнта;
*/
/*
STRING name[11] - прізвище клієнта;
*/
/*
float balance - баланс клієнта.
*/
/*
Сконструювати базовий клас Client з полями:
*/
/*
Закриті елементи класу Client: - відсутні;
*/
/*
Відкриті елементи класу Client:
*/
/*
int K - кількість клієнтів;
*/
/*
client m[100] - масив структур типу client;
*/
/*
void enter() - функція введення даних;
*/
/*
void print() - функція виведення даних;
*/
/*
Сконструювати похідний клас Sort з полями:
*/
/*
Закриті елементи класу Sort:
*/
/*
void swop(int,int) - функція обміну значень.
*/
/*
Відкриті елементи класу Sort:
*/
/*
void sort(int) - функція сортування масиву;
*/
/*
void sort1() - функція сортування масиву
з ручним введенням способу
(до 10 спроб) і виведенням
на екран;
*/
/*
Доступ до елементів класу Sort здійснити
за допомогою об'єкта obS та вказівника на
об'єкт *obSp.
*/
*****
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <iomanip.h>
#include <process.h>
#define kk 3 // Кількість банківських клієнтів
```

```

typedef char STRING;    // Тип даних STRING

//Оголошення структури client
struct client
{
    int number;          // Номер рахунку клієнта
    STRING name[11];     // Прізвище клієнта
    float balance;       // Баланс клієнта
};

//Оголошення базового класу Client
class Client
{
public:                  // Відкриті елементи класу Client
    Client();           // Конструктор класу Client
    ~Client();          // Деструктор класу Client
    int K;              // Кількість банківських клієнтів
    client m[100];      // Масив структур типу client
    void enter();        // Функція введення даних
    void print();        // Функція виведення даних
};

//Оголошення класу Sort -нащадка класу Client
class Sort:public Client
{
void swop(int,int);     // Функція, яка забезпечує обмін значень

public:                // Відкриті елементи класу Sort
    Sort();            // Конструктор
    ~Sort();           // Деструктор
    void sort(int);     // Сортування масиву
    void sort1();       // Сортування масиву з ручним введенням
                        // способу (до 10 спроб) і виведенням на екран
};

//Опис функцій класу Client

// Опис конструкторів та деструкторів
Client::Client() {K=kk;}
Client::~Client() {cout<<endl;} // звільнення буфера пам'яті

Sort::Sort():Client() {}
Sort::~Sort() {}

//Опис функцій класу Sort

// Функція обміну значеннями
void Sort::swop(int i, int j)
{
    client c=m[i];
    m[i]=m[j];
    m[j]=c;
}

// Функція введення даних

```

```

void Client::enter()
{for (int i=0; i<K; i++)
{cout<<endl;
cout<<" Введіть номер рахунку клієнта: ";
cin>>m[i].number;
cout<<" Введіть прізвище клієнта: ";
cin>>m[i].name;
cout<<" Введіть баланс клієнта: ";
cin>>m[i].balance;
};
}

// Функція виведення даних
void Client::print()
{for (int i=0; i<K; i++)
{cout<<"\n Запис N"<<setw(3)<<i+1
<<"\n number ="<<setw(4)<<m[i].number
<<"\n name ="<<setw(11)<<m[i].name
<<"\n balance="<<setw(10)<<setprecision(3)
<<m[i].balance<<endl;
}
}

// Функція сортування даних
void Sort::sort(int ind)
{for (int i=0; i<K-1; i++)
{for (int j=i+1; j<K; j++)
{switch(ind)
{case 1: if (m[i].number>m[j].number) swop(i,j);break;
case 2: if (strcmp(m[i].name,m[j].name) >0)
swop(i,j);break;
case 3: if (m[i].balance>m[j].balance)
swop(i,j);break;
default: cout<<"\n Введено помилковий параметр"<<endl;
goto mm;
};
};
};
mm: ;
}

// Функція сортування даних з ручним введенням параметра сортування
void Sort::sort1()
{clrscr();
for (int i=0; i<10; i++)
{cout<<"\n Виберіть параметр сортування: \n";
cout<<"\n 1 - за номером рахунку";
cout<<"\n 2 - за ім'ям";
cout<<"\n 3 - за величиною балансу";
cout<<"\n 5 - Вихід \n";

int ind;
cin>>ind;
}
}

```



```

        if ( ind == 5) exit(1);

        cout<<"\n Результат сортування за параметром "<<ind<<" : \n";
        sort(ind);
        print();
    };
}

int main()
{clrscr();

    Sort obS, *obSp;
    obSp=&obS;
    obS.enter();
    clrscr();

    obSp->print();
    getch();
    clrscr();

    obSp->sort(2);
    obSp->print();
    getch();
    clrscr();

    obSp->sort1();
    obSp->print();
    getch();
    clrscr();

    return 0;
}

```

Результати тестування програми:

Введіть номер рахунку клієнта:	12345
Введіть прізвище клієнта:	Петренко
Введіть баланс клієнта:	34.65
Введіть номер рахунку клієнта:	23456
Введіть прізвище клієнта:	Тарасенко
Введіть баланс клієнта:	65.65
Введіть номер рахунку клієнта:	13579
Введіть прізвище клієнта:	Катренко
Введіть баланс клієнта:	45.54

```

Запис N 1
number =12345
name   =   Петренко
balance=      34.65

```

```

Запис N 2

```

```
number =23456
name    = Тарасенко
balance=      65.65
```

```
Запис N 3
number =13579
name    = Катренко
balance=      45.56
```

Результат роботи функції sort(2):
яка сортує масив структур за параметром 2 (за ім'ям):

```
Запис N 1
number =13579
name    = Катренко
balance=      45.56
```

```
Запис N 2
number =12345
name    = Петренко
balance=      34.65
```

```
Запис N 3
number =23456
name    = Тарасенко
balance=      65.65
```

Результат роботи функції sort1()
Вибір режиму сортування здійснюється вручну
Для продовження натисніть яку-небудь клавішу!

Виберіть параметр сортування:

- 1 - за номером рахунку
 - 2 - за ім'ям
 - 3 - за величиною балансу
 - 5 - Вихід - ESC
- 1

Результат сортування за параметром 1

```
Запис N 1
number =12345
name    = Петренко
balance=      34.65
```

```
Запис N 2
number =13579
name    = Катренко
balance=      45.56
```

```
Запис N 3
number =23456
```

```
name    =   Тарасенко  
balance=      65.65
```

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Створити , відлагодити та протестувати програму, у якій розробити базовий та похідні класи для обробки масиву структур, створеного у лабораторній роботі №22. Передбачити, що базовий клас виконуватиме функції введення масиву структур, а похідний клас – функції сортування масиву структур.

ЛАБОРАТОРНА РОБОТА №27: “КОНСТРУЮВАННЯ КЛАСІВ З ВИКОРИСТАННЯМ ДРУЖНІХ ФУНКЦІЙ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику конструювання дружніх та віртуальних функцій засобами мови C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика конструювання дружніх та віртуальних функцій засобами мови C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_27_01.

```
/*
 *
 * Дружні функції.
 * Завдання. Сформувані:
 * базовий клас Client та клас-нащадок Sort.
 * Для доступу до закритих елементів класів
 * утворити такі функції:
 * int funCF(Client &obC) - доступ до obc0 -
 * закритого елемента-об'єкта obC класу Client;
 * void funCSF(Client &obC, Sort &obS) -
 * доступ до об'єкта obc0 - закритого елемента
 * класу Client та до функції swop -
 * закритого елемента класу Sort;
 * void Client::call(Sort &ob, int i, int j)
 * функцію-елемент класу Client, дружню до класу Sort.
 */
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <iomanip.h>
#include <process.h>
#include <math.h>

typedef char STRING;    // Оголошення типу STRING
class Sort;             // Посилання

//Оголошення структури client
struct client
{
    int number;          // Номер рахунку клієнта
    STRING name[11];     // Прізвище клієнта
    float balance;       // Баланс клієнта
};

//Оголошення базового класу Client
class Client
{
    client obc0;          // Структурний об'єкт типу client

public:
    int K;                // Кількість клієнтів
};
```

```

Client(int kk) {K=kk;} // Конструктор класу Client
~Client() {}          // Деструктор класу Client
client m[100];        // Масив структур типу client
void enter();          // Функція введення даних
void print();          // Функція виведення даних

// Прототип функції, дружньої до Client
friend int funCF(Client &);

// Прототип функції, дружньої до Client та Sort
friend void funCSF(Client &, Sort &);

// Функція-елемент класу Client, дружня до Sort
void call(Sort &, int,int);

}; // Кінець оголошення класу Client

//Оголошення класу Sort -нащадка класу Client
class Sort:public Client
{void swop(int,int); // Функція, яка забезпечує обмін значень
public:              // Відкриті елементи класу Sort
Sort(int kk=3):Client(kk) {} // Конструктор класу Sort
~Sort(){}            // Деструктор класу Sort
void sort(int);      // Сортування масиву

// Прототип функції, дружньої до Client та Sort
friend void funCSF(Client &,Sort &);

// Функція-елемент класу Client, дружня до Sort
friend void Client::call(Sort &, int,int);

}; // Кінець оголошення класу Sort

//Опис функцій класу Client

void Client::enter()
{for (int i=0; i<K; i++)
{cout<<endl;
cout<<" Введіть номер рахунку клієнта: "; cin>>m[i].number;
cout<<" Введіть прізвище клієнта: "; cin>>m[i].name;
cout<<" Введіть баланс клієнта: "; cin>>m[i].balance;
};
}

void Client::print()
{for (int i=0; i<K; i++)
{cout<<"\n Структура N"<<setw(3)<<i+1<<": "
<<"\n number ="<<setw(4)<<m[i].number
<<"\n name ="<<setw(11)<<m[i].name
<<"\n balance="<<setw(10)<<setprecision(3)<<m[i].balance<<endl;
}
}
}

```

```

//Опис функцій класу Sort

void Sort::swop(int i, int j)
{client c=m[i];
 m[i]=m[j];
 m[j]=c;
}

void Sort::sort(int ind)
{for (int i=0; i<K-1; i++)
 {for (int j=i+1; j<K; j++)
  {switch(ind)
   {case 1:  if (m[i].number>m[j].number) swop(i,j);break;
    case 2:  if (strcmp(m[i].name,m[j].name) >0)
              swop(i,j);break;
    case 3:  if (m[i].balance>m[j].balance) swop(i,j);break;
    default: cout<<"\n Введено помилковий параметр"<<endl;
              goto mm;
   };
  };
};
mm: ;
}

// Опис дружніх функцій

// Доступ до obc0 - закритого елемента об'єкта obC класу Client
// Функція забезпечує:
// введення значень полів однієї структури obc0 об'єкта obC
// збереження сформованої структури в елементі масиву m[0]
// виведення значень полів однієї структури obc0 об'єкта obC
// перевірку платоспроможності клієнта
//
int funcF(Client &obC)
{cout<<"\n Послідовно введіть номер рахунку, прізвище, баланс:
  \n";
 cin>>obC.obc0.number>>obC.obc0.name>>obC.obc0.balance;
 obC.m[0]=obC.obc0;
 obC.print();

 int p;
 // Перевірка платоспроможності
 if (obC.m[0].balance<0)
  p=-1;
 else if (obC.m[0].balance==0)
  p=0;
 else p=1;
 return p;
}

// Доступ до об'єкта obc0 - закритого елемента класу Client
// та до функції swop - закритого елемента класу Sort
// Функція забезпечує:

```

```

// введення значень полів K структур obc0 об'єкта obC
// i їх збереження у масиві m[j] об'єкта obS
void funCSF(Client &obC, Sort &obS)
{for (int j=0;j<obS.K;j++)
    {cout<<"\n Послідовно введіть номер рахунку, прізвище, баланс:
      \n";
      cin>>obC.obc0.number>>obC.obc0.name>>obC.obc0.balance;
      obS.m[j]=obC.obc0;
    };
    clrscr();
    cout<<"\n Значення полів структур об'єкта obS до обміну:\n";
    obS.print();
    getch();
    clrscr();
    cout<<"\n Значення полів структур об'єкта obS після обміну:\n";
    obS.swop(0,obS.K-1);
    obS.print();
    getch();
}

// Функція-елемент класу Client
// Ця функція дружна до класу Sort
// Функція забезпечує:
// обмін значень елементів масиву m[i] та m[j]
// (структури об'єкта ob);
// виведення на друк полів структури об'єкта ob після цього обміну
void Client::call(Sort &ob, int i, int j)
{ob.swop(i,j);
  print();
  getch();
}

int main()
{Client obC(1);
  clrscr();
  // Виклик дружньої функції: аргумент-об'єкт класу Client
  cout<<"\n Демонстрація роботи дружньої функції funCF(obC):\n";
  if (funCF(obC)<0)
    cout<<"\n Клієнт-боржник \n";
  else
    cout<<"\n Клієнт-платоспроможний \n";
  getch();
  clrscr();

  Sort obS; // За замовчуванням аргумент дорівнює 3
  cout<<"\n Демонстрація роботи дружньої функції
        funCSF(obC,obS):\n";
  funCSF(obC,obS);
  cout<<endl;
  getch();
  clrscr();
}

```

```
// Виклик дружньої функції - елементу класу Client
cout<<"\n Демонстрація роботи дружньої функції
      obS.call(obS,0,obS.K-1):\n";
obS.call(obS,0,obS.K-1);
return 0;
}
```

Результати тестування програми:

Демонстрація роботи дружньої функції funCF(obC):

Послідовно введіть номер рахунку, прізвище, баланс:
12345 Петренко 34.65

Структура N 1:
number =12345
name = Петренко
balance= 34.65

Клієнт-платоспроможний

Демонстрація роботи дружньої функції funCSF(obC,obS):

Послідовно введіть номер рахунку, прізвище, баланс:
12345 Петренко 34.65

Послідовно введіть номер рахунку, прізвище, баланс:
23456 Тарасенко 65.65

Послідовно введіть номер рахунку, прізвище, баланс:
13579 Катренко 45.56

Значення полів структур об'єкта obS до обміну:

Запис N 1
number =12345
name = Петренко
balance= 34.65

Запис N 2
number =23456
name = Тарасенко
balance= 65.65

Запис N 3
number =13579
name = Катренко
balance= 45.56

Значення полів структур об'єкта obS після обміну:

Запис N 1
number =13579


```
name    =    Катренко
balance=    45.56
```

```
Запис N 2
number  =23456
name    =    Тарасенко
balance=    65.65
```

```
Запис N 3
number  =12345
name    =    Петренко
balance=    34.65
```

Демонстрація роботи дружньої функції `obS.call(obS,0,obS.K-1):`

```
Запис N 1
number  =12345
name    =    Петренко
balance=    34.65
```

```
Запис N 2
number  =23456
name    =    Тарасенко
balance=    65.65
```

```
Запис N 3
number  =13579
name    =    Катренко
balance=    45.56
```

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

1. Створити абстрактний клас Фігура з віртуальними методами обчислення площі та периметру. Створити похідні класи: прямокутник, круг, трапеція зі своїми функціями площі та периметру. Самостійно визначити, які поля необхідні, які з них можна задати в базовому класі, а які – в похідних.

2. Створити абстрактний базовий клас Тіло з віртуальними функціями обчислення площі поверхні та об'єму. Створити похідні класи: паралелепіпед, куля зі своїми функціями обчислення площі та об'єму.

3. Створити абстрактний клас Валюта для роботи з грошима. Визначити віртуальні функції переведення в гривні і виведення на екран. Реалізувати похідні класи долар та євро зі своїми функціями переведення та виведення на екран.

4. Створити базовий клас Трикутник з віртуальними функціями обчислення площі та периметра. Поля даних повинні включати дві сторони і кут між ними. Визначити класи-нащадки: прямокутний трикутник, рівнобедрений трикутник, рівносторонній трикутник зі своїми методами обчислення площі та периметра.

5. Створити абстрактний базовий клас Корінь з віртуальними методами обчислення коренів рівнянь та виведення на екран. Визначити похідні класи: лінійне рівняння та квадратне рівняння з власними методами знаходження та виведення на екран коренів.

6. Створити абстрактний базовий клас Функція з віртуальними методами обчислення значення функції в заданій точці та виведення результату на екран. Визначити похідні класи: еліпс та гіпербола з власними методами.

7. Створити абстрактний базовий клас Number з віртуальними методами – арифметичними операціями додавання, віднімання, множення, ділення, піднесення до степеня. Визначити похідні класи: ціле та дійсне число.
8. Створити абстрактний базовий клас з віртуальними арифметичними операціями. Реалізувати похідні класи: комплексне число та раціональний дріб.
9. Варіант 9. Створити абстрактний базовий клас з віртуальним методом збільшення на одиницю. Реалізувати похідні класи: дата і час з власними методами збільшення на одиницю.
10. Варіант 10. Створити абстрактний базовий клас прогресія з віртуальними функціями визначення j -го елемента прогресії та суми прогресії. Визначити похідні класи: арифметична та геометрична прогресія з власними відповідними функціями.
11. Варіант 11. Створити абстрактний клас Фігура з віртуальними методами обчислення площі та периметру. Створити похідні класи: прямокутник, круг, трапеція зі своїми функціями площі та периметру. Самостійно визначити, які поля необхідні, які з них можна задати в базовому класі, а які – в похідних.
12. Варіант 12. Створити абстрактний базовий клас Тіло з віртуальними функціями обчислення площі поверхні та об'єму. Створити похідні класи: паралелепіпед, куля зі своїми функціями обчислення площі та об'єму.
13. Варіант 13. Створити абстрактний клас Валюта для роботи з грошима. Визначити віртуальні функції переведення в гривні і виведення на екран. Реалізувати похідні класи долар та євро зі своїми функціями переведення та виведення на екран.
14. Варіант 14. Створити базовий клас Трикутник з віртуальними функціями обчислення площі та периметра. Поля даних повинні включати дві сторони і кут між ними. Визначити класи-нащадки: прямокутний трикутник, рівнобедрений трикутник, рівносторонній трикутник зі своїми методами обчислення площі та периметра.
15. Варіант 15. Створити абстрактний базовий клас Корінь з віртуальними методами обчислення коренів рівнянь та виведення на екран. Визначити похідні класи: лінійне рівняння та квадратне рівняння з власними методами знаходження та виведення на екран коренів.

ЛАБОРАТОРНА РОБОТА №28: “РОБОТА З ОДНОЗВ’ЯЗНИМИ ЛІНІЙНИМИ СПИСКАМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику роботи з однозв’язними списками засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика роботи з однозв’язними списками засобами мови C/C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_28_01.

```
/*
*****
/*
/*      Формування однозв’язного динамічного списку.
/*      Сформувати однозв’язний динамічний список-чергу з даних,
/*      що складаються з індекса та текстового повідомлення.
/*      Передбачити вилучення зі списку елементів з непарними
/*      індексами. Надрукувати список до вилучення і після
/*      вилучення з нього елементів. Завдання виконати мовою C.
/*
*****
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define MES_LEN 200          // Розмірність рядка повідомлення

typedef struct informs       // (Пере)визначення типу даних struct
    informs
{
    int index;               // Поле index - число цілого типу
    char message[MES_LEN];  // Поле message[MES_LEN] - стрічка
    символів
} INFORM;                   // INFORM - назва шаблону структури

// Структура елемента списку:
typedef struct list_elem    // (Пере)визначення типу даних struct
list_elem
{
    INFORM inform;          // Поле inform - типу INFORM
    struct list_elem *next; // Поле *next - вказівник на struct
list_elem
} LEL;                      // LEL - назва шаблону структури

// Прототипи функцій та опис глобальних змінних:
LEL* AddElem(LEL* last);
void PrintList(void);
LEL* FindOddIndex(LEL* start);
LEL* DeleteElem(LEL* pdel, LEL* prev);
void DeleteAllOdd(void);
void FreeList(void);
LEL *list;                  // Глобальний вказівник на початок списку

int main(void)
```

```

{clrscr();
LEL *end=NULL;          // Вказівник на останній елемент списку

// Формування списку:
puts("\t\t Вхідні дані:");
do
    {end=AddElem(end);
    }while (end != NULL);
puts("\n\n\t Введено дані:");

// Друк сформованого списку:
PrintList();

// Вилучення елементів з непарними номерами зі списку:
DeleteAllOdd();

// Друк списку після вилучення елементів:
puts("\n\n\t Список після видалення:");

// Друк скороченого списку:
PrintList();

// Вивільнення динамічної пам'яті:
FreeList();

getch();
return 0;
}

//-----
// Функція приєднання до черги нового елемента
//-----
LEL* AddElem(LEL* last)
{ LEL *pel;          // pel - вказівник на новий елемент
  static int num=1;   // num - номер елемента, що вводиться

  pel=(LEL *)malloc(sizeof(LEL)); // Виділення ДП для елемента

  printf("\n\t%d-й елемент:  індекс - ",num);
  scanf("%d", &pel->inform.index); // Ввід значення індекса

  if (pel->inform.index == 0)      // Ознака кінця введення
  { free(pel);
    return NULL;
  }

  fflush(stdin);                  // Очищення буфера введення

  printf(" повідомлення: ");      // Ввід повідомлення
  gets(pel->inform.message);

  pel->next=NULL;                 // Новий елемент буде останнім у списку

```

```

    if (list == NULL)    // Якщо список порожній
        list=pel;
    else
        last->next=pel;  // Приєднання до останнього у списку

    num++;
    return pel;
}

//-----
// Функція виведення на екран усіх даних списку
//-----
void PrintList(void)
{LEL *pel=list;
  while (pel != NULL)
  {printf("\n%-8d%-70s", pel->inform.index, pel->inform.message);
    pel=pel->next;
  }
}

//-----
// Функція видалення заданого елемента
//-----
LEL* DeleteElem(LEL* pdel, LEL* prev)
{if (pdel==list)        // Видалення першого елемента
  {list=list->next;
   free(pdel);
   return list; // Вказівник на список після видалення
  }

  while (prev->next!=pdel) // Пошук елемента, що передує pdel
    prev=prev->next;
  prev->next=pdel->next;  // Перемикання зв'язків
  free(pdel);

  return prev; // Вказівник на останній перевірений елемент
}

//-----
// Функція пошуку елемента з непарним індексом
//-----
LEL* FindOddIndex(LEL* pel)
{while (pel!=NULL)
  if (pel->inform.index % 2 != 0) // індекс елемента - непарний
    return pel;
  else
    pel=pel->next;
  return NULL; // не знайдено елемента з непарним індексом
}

//-----

```

```

// Функція видалення всіх елементів з непарними індексами
//-----
void DeleteAllOdd(void)
{LEL *pst=list, *pdel;
  while ((pdel = FindOddIndex(pst)) != NULL) // Циклічний пошук
    pst=DeleteElem(pdel, pst); // Видалення елементів списку
}

//-----
// Функція витирання всього списку
//-----
void FreeList(void)
{LEL *pel=list;
  while (pel != NULL)
  {list = list->next;
    free(pel); // Витирання поточного елемента
    pel=list;
  }
}

```

Результати тестування програми:

Вхідні дані:

1-й елемент: індекс - 205
повідомлення: Понизити рівень сигналу

2-й елемент: індекс - 318
повідомлення: Розпочати процес регулювання

3-й елемент: індекс - 202
повідомлення: Підвищити рівень сигналу

4-й елемент: індекс - 765
повідомлення: Аварійна ситуація

5-й елемент: індекс - 381
повідомлення: Перевірка діапазону частот

6-й елемент: індекс - 456
повідомлення: Завершити контроль

7-й елемент: індекс - 0

Введено дані:

205	Понизити рівень сигналу
318	Розпочати процес регулювання
202	Підвищити рівень сигналу
765	Аварійна ситуація
381	Перевірка діапазону частот
456	Завершити контроль

Список після видалення:

318	Розпочати процес регулювання
-----	------------------------------

202 Підвищити рівень сигналу
456 Завершити контроль

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Розв'язати задачу з лабораторної роботи №22 за індивідуальним варіантом представививши дані не у вигляді масиву структур, а у вигляді однозв'язного списку.

ЛАБОРАТОРНА РОБОТА №29: “РОБОТА З ДВОЗВ’ЯЗНИМИ ЛІНІЙНИМИ СПИСКАМИ ЗАСОБАМИ МОВИ C/C++”

МЕТА РОБОТИ

1. Опанувати методику роботи з двозв’язними списками засобами мови C/C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика роботи з двозв’язними списками засобами мови C/C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_29_01.

```
/*
 *
 * Формування двозв’язного динамічного списку.
 * Сформувати двозв’язний динамічний список-чергу з даних,
 * що складаються з індекса та текстового повідомлення.
 * Передбачити вилучення зі списку елементів з непарними
 * індексами. Надрукувати список до вилучення у прямому та
 * зворотному порядку, а також після вилучення з нього
 * елементів з непарними номерами.
 * Завдання виконати мовою C.
 */
/*****
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_LEN 200 // Максимальний розмір повідомлення

typedef struct informs // (Пере)визначення типу даних struct informs
{
    int index; // Поле index - типу int
    char *mes; // Поле *mes - вказівник на char
} DMINF; // DMINF - назва шаблону структури

typedef struct list_elem // (Пере)визначення типу даних struct
list_elem
{
    DMINF inf; // Поле inf - структура за шаблоном з імям
    DMINF
    struct list_elem *next, *prev; // Поля *next, *prev - вказівники
// на структуру
} EL2WL; // EL2WL - назва шаблону структури

// Прототипи функцій:
int InputData(DMINF* inf);
void InsertElem(DMINF data);
void AddElem(EL2WL* pnew, EL2WL* pold);
void PrintList(void);
void PrintReverse(EL2WL* last);
EL2WL* DeleteElem(EL2WL* pdel);
void FreeElemMemo(EL2WL* pel);
```



```

void DeleteAllOdd(void);
void FreeList(EL2WL* start);

// Глобальні вказівники на початок та кінець списку:
EL2WL *list_beg, *list_end;

int main(void)
{clrscr();
  DMINF input;

  // Формування списку:
  puts("\n\t\t Вхідні дані:");
  while (InputData(&input) != 0)
    InsertElem(input);

  // Друк сформованого списку у прямому порядку:
  puts("\n\n\t Сформований список:");
  PrintList();

  // Друк сформованого списку у зворотному порядку:
  puts("\n\n\t Введений список у зворотньому порядку:");
  PrintReverse(list_end);

  // Вилучення елементів з непарними номерами зі списку:
  DeleteAllOdd();

  // Друк скороченого списку:
  puts("\n\n\t Список після видалення:");
  PrintList();

  // Вивільнення динамічної пам'яті:
  FreeList(list_beg);

  getch();
  return 0;
}

//-----
//Функція введення даних елемента списку
//-----
int InputData(DMINF* pinf)
{char buf[MAX_LEN];          // buf - стрічка символів повідомлення
  static int num=0;          // num - значення індекса повідомлення

  printf("\n\t%d-й елемент:  індекс - ", ++num);
  scanf("%d", &pinf->index);
  if (pinf->index != 0)      // Кінець введення - введено індекс 0
    {printf(" Повідомлення: ");
      fflush(stdin);        // Очищення буфера потоку введення
данях
      gets(buf);           //Введення символів повідомлення аж до натискання
Enter

```

```

    pinfo->mes=(char *)malloc(strlen(buf)+1); // Виділення дин.
пам'яті
    strcpy(pinfo->mes,buf);      // Запис рядка повідомлення
}
else
    pinfo->mes=NULL;

return pinfo->index;
}

//-----
//Функція приєднання нового елемента в порядку зростання індексів
//-----
void InsertElem(DMINF data)
{EL2WL *pel, *pins;
    pel=(EL2WL *)malloc(sizeof(EL2WL)); // Виділення дин. пам'яті
    pel->inf=data;                      // Заповнення поля даних
    if (list_beg==NULL)                // Якщо в списку ще не було елементів
    {pel->next=pel->prev=NULL;
        list_beg=list_end=pel; // Єдиний елемент в списку
        return;
    }
    pins=list_beg; // Цикл пошуку позиції вставляння нового
                  // елемента
    while (data.index>pins->inf.index && pins != NULL)
        pins=pins->next;

    AddElem(pel, pins); // Приєднання нового елемента перед *pins
}

//-----
// Функція приєднання до списку елемента
// *pnew перед елементом *pold
//-----
void AddElem(EL2WL* pnew, EL2WL* pold)
{if (pold==list_beg)
    {list_beg->prev=pnew; //Новий елемент стає першим у списку
        pnew->next=list_beg;
        pnew->prev=NULL;
        list_beg=pnew;
        return;
    }
    if (pold!=NULL)
    {pnew->next=pold;      //Новий елемент вводиться у середину
                          //списку
        pnew->prev=pold->prev;
        pold->prev->next=pnew;
        pold->prev=pnew;
    }
    else
        {list_end->next=pnew; //Новий елемент стає останнім у списку

```

```

        pnew->prev=list_end;
        pnew->next=NULL;
        list_end=pnew;
    }
}

//-----
//Функція виведення даних списку у прямому порядку
//-----
void PrintList(void)
{EL2WL* pel=list_beg;
  while (pel != NULL)
    {printf("\n%-8d%-70s", pel->inf.index, pel->inf.mes);
      pel=pel->next;
    }
}

//-----
//Функція (рекурсивна) виведення даних у зворотному порядку
//-----
void PrintReverse(EL2WL* last)
{if (last == NULL) return;
  printf("\n%-8d%-70s", last->inf.index, last->inf.mes);
  PrintReverse(last->prev);
}

//-----
// Функція видалення елемента списку (повертає вказівник на
// наступний за видаленим елемент)
//-----
EL2WL* DeleteElem(EL2WL* pdel)
{if (pdel==list_beg)
  {list_beg=list_beg->next; //Видалення першого елемента списку
   list_beg->prev=NULL;
   FreeElemMemo(pdel);
   return list_beg;
  }
if (pdel==list_end)
  {list_end=list_end->prev; //Видалення останнього елемента
                           //списку

   list_end->next=NULL;
   FreeElemMemo(pdel);
   return NULL;
  }
else
  {EL2WL *pNext; //Видалення серединного елемента
                 // списку

   pdel->next->prev=pdel->prev;
   pnext=pdel->prev->next=pdel->next;
   FreeElemMemo(pdel);
   return pnext;
  }
}

```

```

//-----
//Функція видалення всіх елементів списку з непарними індексами
//-----
void DeleteAllOdd(void)
{EL2WL* pel=list_beg;
  while (pel != NULL)
    if (pel->inf.index % 2 !=0)
      pel=DeleteElem(pel);      //Видалення елемента
    else
      pel=pel->next;
}

//-----
//Функція (рекурсивна) витирання списку
//-----
void FreeList(EL2WL* start)
{if (start == NULL) return;
  list_beg=start->next;
  FreeElemMemo(start); // Витирання початкового елемента
  FreeList(list_beg);  // Рекурсивне продовження
}

//-----
//Функція звільнення динамічної пам'яті, зайнятої елементом списку
//-----

void FreeElemMemo(EL2WL* pel)
{free(pel->inf.mes); // Витирання рядка повідомлення
  free(pel);        // Витирання елемента списку
}

```

Результати тестування програми:

Вхідні дані:

1-й елемент: індекс - 205
повідомлення: Понизити рівень сигналу

2-й елемент: індекс - 318
повідомлення: Розпочати процес регулювання

3-й елемент: індекс - 202
повідомлення: Підвищити рівень сигналу

4-й елемент: індекс - 765
повідомлення: Аварійна ситуація

5-й елемент: індекс - 381
повідомлення: Перевірка діапазону частот

6-й елемент: індекс - 456
повідомлення: Завершити контроль

7-й елемент: індекс - 0

Сформований список:

202	Підвищити рівень сигналу
205	Понизити рівень сигналу
318	Розпочати процес регулювання
381	Перевірка діапазону частот
456	Завершити контроль
765	Аварійна ситуація

Зворотний порядок даних:

765	Аварійна ситуація
456	Завершити контроль
381	Перевірка діапазону частот
318	Розпочати процес регулювання
205	Понизити рівень сигналу
202	Підвищити рівень сигналу

Список після видалення:

202	Підвищити рівень сигналу
318	Розпочати процес регулювання
456	Завершити контроль

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

Завдання 1. Розв'язати задачу з лабораторної роботи №22 за індивідуальним варіантом представививши дані не у вигляді масиву структур, а у вигляді двозв'язного списку.

ЛАБОРАТОРНА РОБОТА №30: “РОБОТА З ФАЙЛАМИ ЗАСОБАМИ МОВИ C++”

МЕТА РОБОТИ

1. Опанувати методику роботи з файлами засобами мови C++.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглядаються питання:

методика роботи з файлами засобами мови C++.

ТИПОВЕ ЗАВДАННЯ

1. Ввести, відлагодити та протестувати програму під назвою lab_work_30_01.

```
/* **** */
/*      Файли послідовного доступу (мова C++).                */
/*      Завдання. В режимі створення файлу і запису в нього    */
/*      значень передбачити:                                    */
/*      1. Створити два файли послідовного доступу             */
/*      "text1.txt" та "text2.txt" засобами мови C++.           */
/*      2. У файлі "text1.txt" зберегти значення елементів     */
/*      масиву чисел:                                           */
/*      a[]={5.8,4.16,3.24,6.07,-2.1,-0.3,1.1,22.17,311.2,0.197}. */
/*      3. У файлі "text2.txt" зберегти значення квадратів     */
/*      елементів заданого масиву з непарними індексами.      */
/*      4. Елементи масиву розмістити в обох файлах у стовпцях. */
/*      5. В режимі зчитування передбачити виведення значень  */
/*      елементів обох файлів на екран дисплея.                */
/* **** */
```

```
#include <iostream.h>
#include <fstream.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <windows.h>
#define N 10
```

```
double x;
int i;
double a[]={5.8,4.16,3.24,6.07,-2.1,-0.3,1.1,22.17,311.2,0.197};
char str[10];
int main () {
    clrscr();
```

```
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
```

```
    //створюємо об'єкт out1 класу ofstream і зв'язуємо його з файлом
    text1.txt
    ofstream out1("text1.txt", ios::out);
    if(!out1.is_open()) { //якщо файл out1 не вдалося відкрити
        cout << "Файл не вдалося відкрити!" << endl;
```

```

        exit(1); //виходимо з програми
    }
    ofstream out2("text2.txt", ios::out);
    if(!out2.is_open()) { //якщо файл out2 не вдалося відкрити
        cout << "Файл не вдалося відкрити!" << endl;
        exit(1); //виходимо з програми
    }
    //Перетворення чисел масиву a[i] до стрічкового вигляду
    //і запис цих стрічок у файл out1 ("text1.txt")
    for ( i=0;i<N;i++) {
        x=a[i]; //
        out1<<gcvt(x,8, str)<<"\n";
        //Перетворення квадратів елементів масиву з непарними
індексами
        //до стрічкового вигляду і запис цих стрічок у файл out2
("text2.txt")
        if (i%2 != 0) {
            x=a[i];
            out2<<gcvt(x*x,8, str)<<"\n";
        }
    }
    out1<<'\n'; //Додавання до файла out1 завершального нуля
    out2<<'\n'; //Додавання до файла out2 завершального нуля
    out1.close(); //Закриття файла out1
    out2.close(); //Закриття файла out2

    ifstream fin1("text1.txt",ios::in); //Відкриваємо об'єкт fin1 класу
ifstream і зв'язуємо його з файлом text.txt
                                                    //відкриваємо цей
об'єкт(файл) для зчитування даних з файла
    if(!fin1.is_open()) { //якщо файл fin1 не вдалося відкрити
        cout << "Файл fin1 не вдалося відкрити!" << endl;
        exit(1); //виходимо з програми
    }
    else { //Зчитування даних з текстового файла fin1 і виведення їх
на екран
        cout<<"\n Робота з файлами послідовного доступу.";
        cout<<"\n Результат зчитування даних з текстового файла
fin1 (\"text1.txt\") "
        <<"\n після запису у цей файл числових даних заданого
масиву:\n\n";
        while (!fin1.eof()) { //поки не досягнуто кінця файла
            fin1>>str; //зчитуємо поточний елемент файла
fin1 в змінну str
            cout<<str<<"\n"; //виводимо на екран поточне слово, а
потім пробіл
        }
    }
    fin1.close(); //Закриття файла out1

    ifstream fin2("text2.txt",ios::in); //Відкриваємо об'єкт fin2 класу
ifstream і зв'язуємо його з файлом text.txt

```

```

//відкриваємо цей
объект(файл) для зчитування даних з файла
    if(!fin2.is_open()) { //якщо файл fin2 не вдалося відкрити
        cout << "Файл fin2 не вдалося відкрити!" << endl;
        exit(1); //виходимо з програми
    }
    else { //Зчитування даних з текстового файл fin2 і виведення їх
на екран
        cout<<"\n Результат зчитування даних з текстового файла
fin2 (\"text2.txt\") "
            <<"\n після запису у цей файл числових даних - "
            <<"\n квадратів елементів масиву з непарними
індексами:\n\n";
        while (!fin2.eof()) { //поки не досягнуто кінця файла fin2
("text2.txt")
            fin2>>str; //зчитуємо поточний елемент файла
fin2 в змінну str
            cout<<str<<"\n"; //виводимо на екран поточне слово, а
потім пробіл
        }
    }
    fin2.close(); //Закриття файла out2
    getch();

    return 0;
}

```

ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

1. Створити текстовий файл LR30.txt, що містить цілі числа. Зчитати дані з файлу і занести парні числа в файл LR30_1.txt, а непарні в файл LR30_2.txt. Визначити яких чисел більше.
2. Створити файл цілих чисел з N компонент. Вивести на екран суму компонентів файлу, які мають непарні індекси і кратні трьом. Вивести на екран вміст файлу.
3. Створити файл цілих чисел з N компонент. Вивести на екран середнє арифметичне компонент файлу, які більші за 10, але менші за 100. Вивести на екран вміст файлу.
4. Створити файл цілих чисел з N компонент. Вивести на екран мінімальний компонент файлу. Вивести на екран вміст файлу.
5. Створити файл F цілих чисел, що складається з N компонент. Після цього створити файли G та H. У файл G записати всі компоненти файлу F, піднесені до квадрату, а у файл H елементи файлу G, поділені на 2. Вивести на екран вміст файлів F, G, H.
6. Створити файл цілих чисел, що складається з N компонент. Визначити і вивести на екран максимальне з від'ємних чисел у файлі.
7. Створити файл F цілих чисел, що складається з N компонент. Після цього створити файл G. У файл G записати всі парні і нульові числа з файлу F. Вивести на екран вміст файлів F та G.
8. З клавіатури ввести цілі a та b і занести їх в файл F. Дописати в файл значення їх суми, різниці, добутку та цілочисельного залишку від ділення a на b.
9. Створити файл F цілих чисел, що складається з N компонент. Переглянути вміст файлу і числа, кратні 2 записати в файл F2, кратні 3 – в файл F3, кратні 5 – в файл F5, кратні 10 – в файл F10 (використайте оператор case, нічого якщо одне й те ж число буде записане в кілька файлів). Виведіть на екран вміст всіх файлів.
10. Створити текстовий файл L, що містить цілі числа. Зчитати дані з файлу і занести парні числа в файл L1, а непарні в файл L2. Визначити яких чисел більше.
11. Створити файл цілих чисел з N компонент. Вивести на екран суму компонентів файлу, які мають непарні індекси і кратні трьом. Вивести на екран вміст файлу.

12. Створити файл цілих чисел з N компонент. Вивести на екран середнє арифметичне компонент файлу, які більші за 2, але менші за 50. Вивести на екран вміст файлу.

13. Створити файл цілих чисел з N компонент. Вивести на екран мінімальний компонент файлу. Вивести на екран вміст файлу.

14. Створити файл F цілих чисел, що складається з N компонент. Після цього створити файли G та H . У файл G записати всі компоненти файлу F , піднесені до квадрату, а у файл H елементи файлу G , поділені на 8. Вивести на екран вміст файлів F , G , H .

15. Створити файл цілих чисел, що складається з N компонент. Визначити і вивести на екран середнє арифметичне від'ємних чисел у файлі.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна література

1. Дейтел Х.М., Дейтел П.Дж. Как программировать на C++. –М.: ООО "Бином-Пресс", 2010. – 1456с.
2. Дейтел Х.М., Дейтел П.Дж. Как программировать на C. –М.: ООО "Бином-Пресс", 2006. – 912с.
3. Б. Страуструп. Программирование. Принципы и практика использования C++. — М. : ООО “Изд. Дом Вильямс”, 2011. — 1248 с. : ил
4. Архангельский А.Я. Программирование в C++ Builder – М.: ООО "Бином-Пресс", 2010. – 896 с.
5. Архангельский А.Я. Приемы программирования в C++ Builder 6 и 2006. – М.: ООО "Бином-Пресс", 2010. – 992с.
6. Архангельский А. Я. Разработка прикладных программ для Windows в C++Builder 5.– М.: Бином-Пресс 2000. – 256 с.

Додаткова література

7. Програмування: Конспект лекцій для здобувачів першого (бакалаврського) рівня освітньо-професійної програм «Комп’ютерна інженерія» та «Кібербезпека» галузь знань 12 «Інформаційні технології» спеціальностей 123 Комп’ютерна інженерія та 125 Кібербезпека денної та заочної форм навчання/ уклад. П.А. Пех. – Луцьк : Луцький НТУ, 2020. – 202 с. формат А4
8. Програмування: Методичні вказівки до лабораторних занять для здобувачів першого (бакалаврського) рівня освітньо-професійної програм «Комп’ютерна інженерія» та «Кібербезпека» галузь знань 12 «Інформаційні технології» спеціальностей 123 Комп’ютерна інженерія та 125 Кібербезпека денної та заочної форм навчання/ уклад. П.А. Пех. – Луцьк : Луцький НТУ, 2020. – 173 с. формат А4
9. Бобровский С. Самоучитель программирования на языке C++ в системе Borland Builder 5.0. – М.: ДЕСС КОМ, I-PRESS, 2000. – 272 с.
10. 13. Глинський Я.М., Анохін В.Є., Рязська В.А. C++ і C++Builder -Львів:СПД Глинський, 2011. – 192 с.
11. В. Н. Зорянский, С. В. Глушаков, С. Н. Хоменко Программирование в среде Borland C++ Builder 6. –Х.: Фолио, 2003. – 508с.
12. Культин Никита. C++ Builder в задачах и примерах. – БХВ-Петербург. –2005. – 336с.
13. Культин Н. Б. Самоучитель C++ Builder.С.Пб: БХВ-Петербург, 2004. –320 с.
14. Федоренко Юрий. Алгоритмы и программы на C++ Builde» М.:– ДМК Пресс.- 2010. – 542 с.
15. Хвищун І.О. Програмування та математичне моделювання. –К.: ВидавничийДім "Ін Юре", 2007. – 544 с.
16. <http://www.cplusplus.com>
17. <http://www.cplusplus.com/ref>

ДЛЯ ПОДАТОК

Навчальне видання

Автори-упорядники:
Пех Петро Антонович,
Лавренчук Світлана Василівна,
Делявський Михайло Володимирович,
Гринюк Сергій Васильович

Лабораторний практикум з програмування мовою C/C++

*Навчальний посібник для студентів технічних спеціальностей
закладів вищої освіти I–IV рівнів акредитації*

Друкується в авторській редакції

Формат 60x84 ¹/₈. Обсяг 26,50 ум. друк. арк., 26,37 обл.-вид. арк.
Наклад 300 пр. Зам. 70. Видавець і виготовлювач – Вежа-Друк
(м. Луцьк, вул. Шопена, 12, тел. (0332) 29-90-65).
Свідоцтво Держ. комітету телебачення та радіомовлення України
ДК № 4607 від 30.08.2013 р.