

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеська національна академія зв'язку ім. О. С. Попова

Кафедра інформаційних технологій

ОСНОВИ ПРОГРАМУВАННЯ

Частина 1

БАЗОВІ АЛГОРИТМИ

**Методичні вказівки до лабораторних і практичних робіт
для студентів напрямку 6.050103 – Програмна інженерія**

Одеса 2014

Укладачі: Трофименко О.Г., Прокоп Ю.В., Швайко І.Г., Буката Л.М.

Рецензент: к.т.н., доцент Флейта Ю.В.

У методичних вказівках розглядається робота в популярному інтегрованому середовищі розробки програмного забезпечення Visual C++. Призначено для студентів при підготовці до лабораторних і практичних занять з дисципліни "Основи програмування". Оскільки ця дисципліна передбачає значну кількість лабораторних і практичних занять, методичний матеріал для їх підготовки було розбито на декілька частин. У запропонованій Вам першій частині розглянуто засоби Visual C++ для організації обчислень лінійної, розгалуженої і циклічної структур, яким згідно з навчальною програмою присвячені перші дев'ять лабораторних робіт курсу. Кожна із наведених робіт містить короткі теоретичні відомості, приклади створювання програмних проектів засобами Visual C++, питання для самоконтролю та варіанти індивідуальних завдань різних рівнів складності для виконання їх на комп'ютері.

Методичні вказівки будуть корисними студентам спеціальності "Програмна інженерія", які вивчають дисципліну "Основи програмування" для набуття навиків програмування з метою подальшого застосовування цих навиків у власній повсякденній і майбутній професійній діяльності; також стануть у нагоді користувачам персональних комп'ютерів, які бажають навчитися програмувати в середовищі Visual C++.

СХВАЛЕНО

на засіданні кафедри
інформаційних технологій
і рекомендовано до друку.

Протокол № 2 від 01.10.2014 р.

ЗАТВЕРДЖЕНО

методичною радою академії зв'язку.

Протокол № 10/14 від 04.07.2014 р.

Передмова

Популярне інтегроване середовище розробки програмного забезпечення Visual C++ є частиною комплексу Microsoft Visual Studio, а, крім того, воно постачається у вигляді безкоштовного функціонально обмеженого комплексу Visual C++ Express Edition. Стандартизована мова програмування Visual C++ дозволяє розробляти як консольні програми, так і програми з графічним інтерфейсом, у тому числі з підтримкою технології Windows Forms як в рідному, так і в керованому кодах для різних комп'ютерних систем. Поєднання простоти освоєння візуального середовища проектування і підтримка широкого спектра технологій роблять Visual C++ універсальним інструментом створення програмних проектів будь-якого рівня складності як для платформи .NET Framework, так і для виконання в середовищі "чистої" Windows.

У даних вказівках подано короткі теоретичні відомості, приклади створення програмних проектів засобами Visual C++ для організації обчислень лінійної, розгалуженої та циклічної структур, контрольні питання і варіанти індивідуальних завдань до дев'яти лабораторних робіт.

Кожна із запропонованих до виконання лабораторних робіт має завдання різних рівнів складності. Студент сам чи то за вказівкою викладача вибирає завдання того чи іншого рівня складності відповідно до індивідуального варіанта. В подальшому при оцінюванні знань викладач може враховувати рівень складності виконання лабораторної роботи, оптимальність алгоритму програми, вчасність підготовки та виконання роботи.

Перед виконанням лабораторного завдання студентові треба:

- уточнити у викладача індивідуальне завдання;
- вивчити відповідні розділи теоретичного курсу згідно з лекційними записами та навчальною літературою;
- розробити схему алгоритму розв'язання задачі;
- написати тексти програм мовою C++;
- підготувати протокол виконання лабораторної роботи і подати його викладачеві для перевірки.

До виконання лабораторної роботи допускається студент, який має попередньо підготовлений самостійно заповнений протокол лабораторної роботи.

Зміст протоколу лабораторної роботи:

- назва теми і мета лабораторної роботи;
- відповіді на контрольні питання;
- схеми алгоритмів для розв'язання індивідуального завдання;
- тексти програм мовою C++;
- вигляд форми з елементами (у разі створення програмних проектів з формою);
- результати обчислень на комп'ютері.

Правильність роботи програми та здобутих результатів перевіряються і оцінюються викладачем.

Структура дисципліни

Дисципліна “Основи програмування” вивчається студентами I курсу на пряму підготовки “Програмна інженерія”.

Метою викладання навчальної дисципліни “Основи програмування” є навчання студентів сучасним методам опрацювання інформації, навикам алгоритмізації та програмування алгоритмічною мовою високого рівня C++, формування знань і навиків створення програмних проектів в об’єктно-орієнтованому середовищі програмування Visual Studio з використанням засобів платформи .NET FrameWork, застосування набутих навиків у процесі навчання і майбутній професійній діяльності, пов’язаній з використанням персональних комп’ютерів у процесі аналізу та керування об’єктами телекомунікацій.

Основними завданнями вивчення дисципліни “Основи програмування” є формування знань і навиків основ конструювання програмного забезпечення, розкриття можливостей використання комп’ютерів для розв’язування прикладних задач у галузі керування об’єктами телекомунікацій, набуття знань для активного застосовування сучасних інформаційних технологій для виконання та оформлення курсових робіт, вивчення можливостей об’єктно-орієнтованого середовища програмування Visual C++ для створення програм алгоритмічною мовою високого рівня C++ і поглиблення знань шляхом самостійної роботи.

У результаті вивчення дисципліни студент повинен:

знати:

- різновиди алгоритмів та основні засоби їхньої побудови;
- елементи алгоритмічної мови програмування C++ (алфавіт мови, типи даних, правила записування арифметичних виразів);
- особливості створення програмних проектів засобами Visual C++;
- оператори мови C++;
- прийоми розробки та реалізації лінійних, розгалужених і циклічних алгоритмів і програм;
- модульний принцип розробки програм;
- засоби програмування базових алгоритмів опрацювання одно- і двовимірних масивів;
- основні прийоми роботи з вказівниками на масиви та динамічною пам’яттю;
- особливості програмного опрацювання символьних даних (рядків);
- прийоми роботи з текстовими та бінарними файлами;

вміти:

- працювати з програмним забезпеченням і файловою системою, проводити найпростіші операції щодо обслуговування комп’ютера, адекватно і обґрунтовано вибирати програмний засіб для розв’язання фахових завдань і здійснювати обмін даними між програмами;
- створювати лінійні, розгалужені та циклічні алгоритми для розв’язування задач; будувати й описувати блок-схеми;

- розробляти програми з лінійною, розгалуженою та циклічною структурами;
- створювати програми для обчислення в циклах скінчених і нескінчених сум, дослідження функцій на певному проміжку з обчисленням таблиць значень і побудовою графіків; створювати програми з вкладеними циклами.
- складати алгоритми та програми мовою C++ для опрацювання елементів векторів і матриць та виконувати їх на комп'ютері, а саме: обчислення елементів вектора або матриці за формулою, сортування елементів масивів, пошук мінімальних (максимальних) значень елементів вектора чи матриці, обчислення сум, добутків і кількості елементів вектора чи матриці за умовою;
- розробляти програмні проекти опрацювання масивів за допомогою вказівників і виділенням динамічної пам'яті;
- розробляти програмні проекти опрацювання символьних даних і рядків;
- створювати й описувати модульні алгоритми та програми з використанням системних і створенням власних функцій та бібліотечних файлів;
- розробляти програмні проекти опрацювання текстових і бінарних файлів.

Лабораторна робота № 1

Знайомство з Visual C++.

Способи введення-виведення даних

Мета роботи: ознайомитися з інтерфейсом Microsoft Visual C++ – інтегрованого середовища розробки програм мовою C++ та набути навиків створення та налагоджування програмних проектів.

Теоретичні відомості

1. Введення-виведення даних

Visual C++ підтримує декілька способів введення-виведення даних залежно від типу програмного проекту (консольний додаток чи проект Windows Forms) та використовуваних команд.

Введення-виведення в стилі C++

Для введення-виведення даних у C++ найчастіше використовуються потокові команди **cin**>> (вхідний потік) та **cout**<< (вихідний потік) бібліотеки **iostream.h**, наприклад:

```
cout<< "Введіть число: ";
cin>> x;
cout<< "Квадрат цього числа: " << x*x << endl;
```

Перша з команда **cout**<< виведе рядок символів і залишить курсор на тому ж самому рядку. При цьому символи рядка виводитимуться по одному, доки не зустрінеться завершальний символ **'\0'**. Друга команда дозволить ввести значення змінної **x**. Третя команда сформує рядок, поряд виведе числове значення та перемістить курсор на новий рядок (маніпулятор **endl**).

До речі, крім маніпулятора **endl**, новий рядок можна сформувати виведенням символу **'\n'**. Ще одним часто використовуваним спеціальним символом для кращого вигляду виведених даних є символ табуляції **'\t'**, який формує відступ.

Зауважимо, що при виведенні тексту для коректного відображення літер кирилиці слід застосувати команду **setlocale(0, ".1251")** (або **setlocale(LC_ALL, "Russian")**). Виведення замість літер кирилиці усілякої абракадабри спричинено тим, що Visual Studio в консольних додатках використовує для набраного тексту кодування Windows 1251, а для введенного тексту – кодування DOS. Коректно відобразити введенний за допомогою **cin**>> текст дозволить команда **setlocale(LC_ALL, ".OCP")**, повернувши початкові налаштування кодування.

Доволі зручною є можливість виведення за допомогою **cout**<< чисел не лише в десятковому форматі, а і в шістнадцятковому чи вісімковому, використовуючи модифікатори **dec**, **hex** і **oct** усередині вихідного потоку:

```
setlocale(0, ".1251");
cout<< "Вісімковий:\t\t " << oct << 10 << " " << 255 << endl;
cout<< "Шістнадцятковий:\t"<< uppercase<<hex<<10<<" " <<255<<endl;
cout<< "Десятковий:\t\t " << dec << 10 << " " << 255 << endl;
```

Результатом виконання цих команд будуть рядки:

Вісімковий	12 377
Шістнадцятковий:	A FF
Десятковий:	10 255

Слід зазначити, що використання одного з цих модифікаторів залишиться в силі, допоки чи то програма не завершиться, чи то не буде використано інший модифікатор.

Для форматування даних при виведенні командою `cout<<` можна використовувати модифікатор `setw`, який дозволяє задавати ширину (мінімальну кількість символівних позицій) кожного виведеного числа. При цьому зазначена модифікатором ширина є дійсною лише для одного числа. Для можливості використання модифікаторів слід долучити заголовний файл `iomanip`. Якщо виведене число має меншу, аніж зазначену у модифікаторі `setw` ширину, перед ним будуть виведені пробіли. Так, команди

```
#include <iomanip>
. . . . .
cout<< "x=" << setw(1) << 155 << endl;
cout<< "x=" << setw(3) << 155 << endl;
cout<< "x=" << setw(5) << 155 << endl;
```

сформують рядки у такому вигляді:

```
x=155
x=155
x= 155
```

Тобто в останньому рядку перед значенням 155 виведуться два пробіли, а все число займе п'ять позицій.

При виведенні дійсних чисел, особливо коли йдеться про виведення матриць дійсних чисел, досить доречним є обмеження кількості знаків після десяткової крапки за допомогою маніпулятора **`setprecision(int count)`**:

```
cout << fixed << setprecision(3) << (13.5 / 2) << endl; // 6.750
cout << fixed << setprecision(2) << 24.16425 << endl;   // 24.16
```

Використаний тут маніпулятор `fixed` задає виведення дійсних чисел з рухомою крапкою у форматі з фіксованою крапкою.

Команда `cout<<` дозволяє перенаправляти виведення на пристрій чи до файла за допомогою операторів перепризначення виведення операційної системи. Однак, повідомлення про помилки звичайно недоречно спрямовувати до файла і при цьому не виводити на екран. Для уникнення таких ситуацій існує спеціальний вихідний потік `cerr`, пов'язаний зі стандартним пристроєм помилок. Наприклад, команда

```
cerr<< "Повідомлення про помилку" << endl;
```

сформує відповідне повідомлення на екрані, а операційна система не дозволить перенаправити виведення на інший пристрій чи до файла.

Введення-виведення у стилі C

1) Для форматованого введення-виведення в C існують функції **`scanf`** (форматоване введення даних) та **`printf`** (форматоване виведення даних), які містяться в бібліотеці `stdio.h`. Обидві функції мають схожий формат:

```
scanf(<формат>, <список_змінних>);
printf(<формат>, <список_змінних>);
```

де: *формат* – рядок специфікаторів формату у подвійних лапках. Найбільш поширеними специфікаторами є: %i – для цілих чисел, %f – для дійсних чисел типу float чи double, %s – для рядка символів;
список_змінних – послідовність розділених комами змінних, значення яких вводиться чи виводяться.

Наприклад, функція printf("x= %7.3f\n", x) виведе дійсне число x у заданому форматі з трьома знаками дійсної частини числа після десяткової крапки, а функція scanf("%i %f", &kol, &vart) введе значення цілої змінної kol і дійсної змінної vart.

2) Існує ще одна пара функцій C для введення-виведення – **gets()** і **puts()**, які використовуються лише для рядків. Ці функції мають лише один параметр: текст повідомлення для функції puts і змінну-рядок для функції puts, наприклад:

```
puts("Hello, Dolly");
char s[13];
gets(s);
```

Функція puts(s) виводить рядок s на екран, замінюючи нуль-символ на *Enter*. Функція gets(s) зчитує символи з клавіатури до появи символу переведення рядка *Enter* і записує їх у рядок s (власне символ *Enter* до рядка не долучається, а замість нього записується нуль-символ).

Введення-виведення у стилі .NET Framework

1) Клас **Console** є частиною бібліотек CLR (Common Language Runtime – середовище виконання .NET-додатків) платформи .NET Framework і містить низку методів для введення-виведення потоків даних в консольних додатках. Цей клас належить до простору імен System, і тому слід записати команду:

```
using namespace System;
```

Методи цього простору Read і ReadLine призначені для читання символів зі стандартного потоку введення, а методи Write і WriteLine – для виведення рядка символів у стандартний потік виведення.

Специфікатори форматів:

f, F – формат з фіксованою дійсною крапкою;
d, D – формат Decimal (десяткові цілі числа);
c, C – грошовий формат;
p, P – процентний формат;
e, E – експоненційне представлення дійсних чисел з рухомою крапкою;
g, G – формат General для перетворювання значень типу double до формату з фіксованою чи рухомою крапкою;
n, N – формат Number для перетворення у форму ddd,ddd.dd

Приклади:

```
double x=54.321, y=0.0125; int n=1234;
Console.WriteLine();
Console.WriteLine("n= {0:d}", n );           // n= 1234
Console.WriteLine("n= {0:d5}", n );         // n= 01234
```

```

Console.WriteLine("x={0:f} y={1:f2}", x, y); // x=54.32 y=0.01
Console.WriteLine("Ставка: \t{0:p}", y); // Ставка: 1.25 %
Console.WriteLine("Ціна товару: \t{0,7:c}", x); // Ціна товару: $54.32
Console.WriteLine("y={0:e} x={0:E2}", y, x); // y=1.250000e-002 x=5.43E+001
Console.WriteLine("n= {0:n}", n); // n= 1,234.00

```

Крім цих специфікаторів форматів, можна застосовувати комплексні шаблони, наприклад:

```

Console.WriteLine("x={0:0.##}", x); // x=54.32
Console.WriteLine("Ставка: \t{0:%#.##}", y); // Ставка: % 1.25
Console.WriteLine("y={0:##.##e+0}", y); // y=12.5e-3

```

2) Клас **Convert** простору імен **System** перетворює значення одного базового типу даних до іншого базового типу даних за допомогою того чи іншого методу перетворення. Приміром, для перетворення до типу **double** існує метод **.ToDouble**, до типу **float** – метод **ToSingle**, до типу **int** – метод **ToInt32** тощо.

```

double x = Convert.ToDouble(textBox1->Text);
textBox2->Text = Convert.ToString(x*x);

```

Поширені функції перетворювання числових типів даних в .NET

Введення і виведення даних у програмних проектах відбувається у текстовому вигляді. В середовищі .NET існує ціла низка методів для перетворення одних типів даних на інші.

Тип	Метод	Виконуване перетворення
String	ToDouble()	перетворює тип String на тип double
String	ToInt32()	перетворює тип String на тип int
double	ToString()	перетворює тип double на тип String
double	Parse()	перетворює тип String на тип double
int	ToString()	перетворює тип int на тип String
int	Parse()	перетворює тип String на тип int

Наприклад:

```

int n = 3; double x;
// виведення рядка з перетвореним цілим числом n
Console.WriteLine(n.ToString());
String *s = S"3.14"; // вказівник на рядок символів
x = Double.Parse(s); // перетворення рядка s на дійсне число x

```

2. Послідовність створення програмних проектів у Visual C++

Visual C++ є частиною комплексу Microsoft Visual Studio, яка дозволяє розробляти як консольні програми, так і програми з графічним інтерфейсом. Visual C++ є унікальним серед інших мовних засобів, що надаються середовищем Visual Studio, оскільки він підтримує розробку програм як для виконання в середовищі "чистої" Windows, так і на керованому (managed) C++. Керований¹ програмний код потребує та виконується виключно віртуальною машиною

¹ Слово керований тут відноситься до методу обміну інформацією між програмою і виконавчим середовищем. Воно означає, що в будь-якій точці виконання, виконавче середовище може припинити виконання і отримати інформацію, специфічну для поточного стану.

Common Language Runtime (скорочено CLR – “загальне середовище виконання мов”, яке є компонентом пакета Microsoft .NET Framework), на якій виконуються всі мови платформи .NET Framework.

Далі буде розглянуто такі засоби створення програмних проектів:

- консольного додатка Win32;
- консольного додатка CLR;
- додатка Windows Forms.

Створення консольного додатка Win32

Після запуску Visual Studio відкриється початкова сторінка, на якій слід виконати команду *Создать проект*, що призведе до відкриття діалогового вікна *Создать проект*. Після вибору мови C++ слід обрати вид створюваного програмного проекту Win32 – *Консольное приложение Win32*. Не закриваючи вікно *Создать проект*, слід ввести ім'я програмного проекту, при цьому для вибору теки для зберігання можна скористатися кнопкою *Обзор*. Після цього натиснути кнопку *ОК*, що призведе до появи діалогового вікна майстра додатків Win32, в якому слід просто натиснути кнопку *Готово*. Після цього відкриється вікно срр-файла програмного проекту з автоматично згенерованим порожнім шаблоном головної функції:

```
#include "stdafx.h"
int _tmain(int argc, _TCHAR* argv[])
{ return 0;
}
```

Тепер можна приступити до написання програмного коду.

Як приклад наведемо програмний код обчислення суми двох чисел:

```
#include "stdafx.h"
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{ setlocale(0, ".1251");
  double a,b;
  cout << "Введіть два числа (через пробіл)\n";
  cin >> a >> b;
  cout << "Сума цих чисел - "<< a + b << endl;
  system ("pause>>void");
  return 0;
}
```

Узагальнене тлумачення програмного коду розглянемо дещо нижче, а за-раз наведемо порядок запуску програм на виконання.


Запустити програму можна одним зі способів:

- командою меню *Отладка – Запуск без отладки*;
- натисненням клавіш *Ctrl – F5*.

Результатом роботи програми буде поява вікна консолі з таким текстом (при натисненні будь-якої клавіші вікно закриється):

```
Введіть два числа (через пробіл)
129.5 0.75
Сума цих чисел - 130.25
```

Ще одним способом запуску програми є запуск налагоджувача:

- командою меню *Отладка – Начать отладку*;
- кнопкою  на панелі інструментів;
- натисненням клавіші *F5*.

Відмінність цього способу буде видно лише за відсутності у програмному коді останньої команди, яка організує паузу з очікуванням натиснення будь-якої клавіші. До речі, останню команду `system("pause>>void")` можна змінити парою команд:


```
cin.get(); cin.get();
```

Ще однією суттєвою відмінністю запуску програми у налагоджувачі є можливість встановлення точок зупину, скориставшись клавішею *F9* чи то командою *Отладка – Точка останова*. При встановленні такої точки, після запуску програми на виконання, вона зупиниться на відповідній команді, і можна буде відслідкувати значення тієї чи іншої змінної на певному етапі виконання програми. Найпростішим способом здійснення цього є наведення покажчика миші на змінну, при цьому біля курсора сформується віконце з поточним значенням цієї змінної.

Тепер, як було обіцяно, наведемо узагальнене тлумачення вищенаведеного програмного коду, а більш глибоке розуміння Ви набудете при подальшому вивченні відповідних програмних засобів.

На початку програми для можливості коректного використання операторів введення-виведення даних `cin` та `cout` у програмному коді було долучено бібліотеку `iostream`. Крім того, оскільки ці команди належать простору імен `std`, було прописано використання цього простору – `using namespace std`.

В основній програмі `_tmain()` на початку для можливості коректного відображення символів кирилиці було долучено використання кодування 1251. Далі оголошено та введено значення двох дійсних (тип `double`) змінних `a` та `b`. Після цього виводиться відповідний коментар та викликається функція `nod()`, числовий результат якої виводиться. Наприкінці організовується пауза (`system("pause>>void")`) як очікування натиснення будь-якої клавіші.

Зберегти всі файли програмного проекту можна командою *Файл – Сохранить все*, чи то натисненням клавіш *Ctrl+Shift+S*, чи то кнопкою  на панелі інструментів.

Створення консольного додатка CLR

CLR – це добавка-розширення C++, введена фірмою Microsoft, яка керує виконанням коду, пам'яттю, потоками і роботою з віддаленими комп'ютерами, при цьому забезпечуючи безпеку і надійність виконання коду.

Відмінність послідовності створення цього шаблону програмного проекту від попереднього полягає у використанні команди *Создать проект – CLR – Консольное приложение CLR*. Після зазначення імені проекту та натиснення кнопки ОК з'явиться вікно `cpp`-файла програмного проекту з таким автоматично згенерованим шаблоном головної функції:

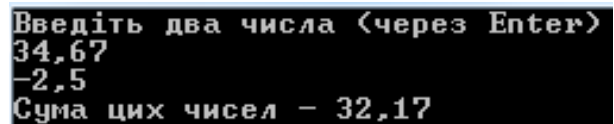
```
#include "stdafx.h"
using namespace System;
```

```
int main(array<System::String ^> ^args)
{
    Console::WriteLine(L"Здравствуй, мир!");
    return 0;
}
```

У цій функції вже існує долучення простору імен System та команда виведення "вітального" рядка WriteLine(). Тобто для введення-виведення даних у CLR-проектах можна використовувати як команди C++ (див. програмний код на стор. 8) так і методи простору імен System, а саме: ReadLine(), WriteLine() для читання та виведення рядка символів та ін. (див. стор. 8 ... 9). Отже, у такому програмному проекті можна реалізувати як програмний код, наведений на стор. 10, так і модифікувати його до такого вигляду:

```
#include "stdafx.h"
using namespace System;

int main(array<System::String ^> ^args)
{ double a,b;
  Console::Write("Введіть два числа (через Enter)\n");
  a = Convert::ToDouble(Console::ReadLine());
  b = Convert::ToDouble(Console::ReadLine());
  Console::WriteLine("Сума цих чисел - {0:f}", a+b);
  Console::Read();
  return 0;
}
```



```
Введіть два числа (через Enter)
34,67
-2,5
Сума цих чисел - 32,17
```

Створення додатка Windows Forms

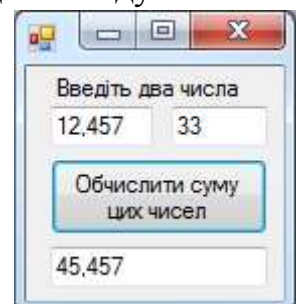
Для створення програмного проекту з формою слід запустити Visual Studio та виконати команду *Создать проект / Visual C++ / CLR / Приложение Windows Forms*. Не закриваючи вікно *Создать проект*, натиснути кнопку *Обзор*, вибрати теку для зберігання та задати ім'я програмного проекту. Натиснути кнопку *OK* та приступити до створення форми і написання програмного коду.

Для налаштування форми слід на ній розмістити необхідні елементи, наприклад: три елементи textBox, один button та один label. Також треба задати значення властивості Text елементів label1 та button1, увівши в ній відповідний текст. Після подвійного клацання по командній кнопці button1 система сформує порожній шаблон функції button1_Click:

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
}
```

Після введення програмного коду шаблон функції набуде вигляду:

```
private: System::Void button1_Click(System::Object^ sender,
                                     System::EventArgs^ e)
{ double a, b;
  a = Convert::ToDouble(textBox1->Text);
  b = Convert::ToDouble(textBox2->Text);
  textBox3->Text = Convert::ToString(a+b);
}
```



Питання та завдання для самоконтролю

- 1) Назвіть різні засоби створювання програмних проектів у Visual C++.
- 2) Що таке консольна програма?
- 3) Назвіть послідовність створювання консольного проекту в Visual C++.
- 4) Які функції введення-виведення даних у консольному режимі Вам відомі?
- 5) За допомогою якої директиви до програми долучають бібліотечні модулі (заголовні файли)?
- 6) Які заголовні файли слід долучити для використання функцій введення-виведення?
- 7) В які способи можна запустити проект на виконання?
- 8) В який спосіб можна зберегти програмний проект?

Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
 - 2) У протоколі лабораторної роботи записати програми мовою C++ для розв'язання завдань.
 - 3) Засобами Visual C++ створити на комп'ютері три різні програмні проекти для визначення суми двох чисел:
 - у вигляді консольного додатка Win32;
 - у вигляді консольного додатка CLR;
 - у вигляді додатка Windows Forms.
- У кожному із програмних проектів слід використовувати відповідні програмні засоби.

Лабораторна робота № 2

Програмування лінійних алгоритмів

Мета роботи: набути практичних навиків записування арифметичних виразів мовою C++ і створення програмних проектів лінійної структури в Visual C++.

Теоретичні відомості

1. Різновиди алгоритмів. Лінійні алгоритми (послідовності)

Базові алгоритми організації обчислень поділяють на три основні види:

- лінійні (послідовності);
- розгалужені;
- циклічні.

Переважно вони є окремими частинами обчислювального процесу, тоді як загальний обчислювальний процес має складнішу (комбіновану) структуру. Здебільшого при написанні програми ці базові алгоритми поєднуються в такій структурі.

Лінійним називається алгоритм, в якому всі дії, від першої до останньої, виконуються послідовно у порядку їхнього запису. Іноді таку структуру називають просто *послідовністю*. Типовим прикладом такого алгоритму є стандартна обчислювальна схема, яка складається з трьох етапів:

- введення початкових даних;
- обчислення за формулами;
- виведення результату.

2. Типи даних C++

У програмі мовою C++ усі змінні мають бути оголошеними, тобто для кожної змінної має бути зазначено її тип. На відміну від інших мов, у C++ *задавати тип змінної можна в будь-якому місці програми до її використання*. При оголошенні змінної для неї резервується ділянка пам'яті, розмір якої залежить від типу змінної.

Тип змінної – вид і розмір даних, які змінна може зберігати. Кожен тип даних зберігається й опрацьовується за певними правилами. Слід зауважити, що розмір одного й того самого типу даних може відрізнятися на комп'ютерах різних платформ, а також залежить від налагоджень компілятора.

Усі типи мови C++ розподіляють на дві групи: основні типи та структуровані. Список основних типів даних C++ із зазначенням діапазону та прикладами можливих значень змінних наведено в табл. 2.1.

До *основних (базових)* типів можна віднести `char`, `int`, `float` та `double`, а також їхні варіанти зі специфікаторами `short` (короткий), `long` (довгий), `signed` (зі знаком) та `unsigned` (без знаку).

Структуровані (похідні) типи базуються на основних, до них належать масиви будь-яких типів, вказівники, функції, класи, файли, структури, об'єднання, перерахування тощо.

Таблиця 2.1

Основні типи даних C++

Тип	Назва	Розмір, байт	Діапазон	Приклади можливих значень	Типи чисел
<code>char</code>	символьний (знаковий)	1	–128...127	'a', '\n', '9'	цілі
<code>unsigned char</code>	беззнаковий символьний	1	0...255	1, 233	
<code>short</code>	короткий цілий	2	–32 768...32 767	1, 153, –349	
<code>unsigned short</code>	беззнаковий короткий	2	0...65 535	0, 4, 65 000	
<code>int</code>	цілий (знаковий)	4*	–2 147 483 648... ...2 147 483 647	–30 000, 0, 690	
<code>unsigned int</code>	беззнаковий цілий	4	0...4 294 967 295	2 348, 60 864	
<code>long</code>	цілий (знаковий)	4	–2 147 483 648... ...2 147 483 647	–30 000, 0, 690	дійсні
<code>float</code>	дійсний одинарної точності	4	$3.4 \cdot 10^{-38}$... $3.4 \cdot 10^{38}$	3.23, –0.2 100.23, 12,	
<code>double</code>	дійсний подвійної точності	8	$1.7 \cdot 10^{-308}$ $1.7 \cdot 10^{308}$	–0.947, 0.0001,	
<code>long double</code>	довгий дійсний	10	$3.4 \cdot 10^{-4932}$ $1.1 \cdot 10^{4932}$	6.34e–3, 4e5	
<code>bool</code>	логічний	1	false чи true	false(0), true(>=1)	
<code>enum</code>	перераховний	2 або 4			
<code>void</code>	порожній, без значення				
* – залежно від налагоджень компілятора і апаратних характеристик тип <code>int</code> може мати 4 або 2 байта.					

Для подання *цілих чисел* використовують типи `char`, `short`, `int`, `long`. Специфікатор `unsigned` застосовують при роботі з додатними числами (без знака), а специфікатор `signed` – для яких завгодно чисел як додатних, так і від’ємних. За замовчуванням призначається знаковий тип, а тому специфікатор `signed` зазначати необов’язково.

Типи `float` і `double` визначають *дійсні змінні* розміром у 32 і 64 біти відповідно, а `long double` – 80 бітів. В C++ для відокремлення цілої частини числа від дійсної застосовується десяткова крапка. Окрім звичної форми, дійсні константи можна записувати у формі з рухомою крапкою. Наприклад: 2.38e3 (яке дорівнює $2.38 \cdot 10^3 = 2380$), 3.61e–4 (яке дорівнює $3.61 \cdot 10^{-4} = 0.000361$). Число перед символом “e” називається *мантисою*, а після символу “e” – *порядком*, тобто замість основи 10 використовується літера “e”, після якої ставиться показник степеня, наприклад: 1e3 ($1 \cdot 10^3 = 1000$), –2.7e–4 ($-2.7 \cdot 10^{-4} = 0.00027$).

Отже, на базових типах (`char`, `int`, `float`, `double`) будується решта типів даних за допомогою специфікаторів:

`signed` (знаковий – за замовчуванням) та `unsigned` (беззнаковий) – до цілих типів `char` та `int`;

`long` (довгий), `short` (короткий) – до типів `double` та `int`.

Наприклад:

```
unsigned int n;  
int c;           // Інтерпретується як signed int c  
short a;         // Інтерпретується як signed short int a  
unsigned d;      // Інтерпретується як unsigned int d  
signed f;        // Інтерпретується як signed int f
```

При оголошуванні змінних їм можна присвоювати початкові значення, які надалі може бути змінено. У прикладі

```
int i, j = 1, q = 0xFFFF;
```

ціла змінна `i` не є ініціалізована, `j` – ініціалізована значенням 1, `q` – ініціалізована шістнадцятковим значенням `0xFFFF` (десятькове 4095).

При ініціалізації змінних їм можна присвоювати арифметичні вирази:

```
long MB=1024*1024;
```

Наведемо ще кілька прикладів оголошення змінних:

```
char tol='a';    // Символьна змінна tol ініціалізується символом 'a'  
char x, c='\n';  // x не є ініціалізований, c – ініціалізований символом <Enter>  
char *s="Одеса"; // Рядок символів
```

Якщо при оголошенні числові змінні не ініціалізовано, то їхні значення є невизначені (випадкові).

Тип `void` (порожній, без значення) використовують для зазначення типу функцій, які не повертають жодного результату, а всі дії та обчислення виконуються всередині цих функцій. Також цей тип застосовують для зазначення порожнього списку аргументів функції.

Для визначення розміру пам'яті, займаної змінною, існує операція `sizeof()`, яка повертає значення довжини зазначеного типу, наприклад:

```
a = sizeof(int);           // a = 4  
b = sizeof(long double);  // b = 10
```

3. Константи

Константа – величина, яка не змінюється упродовж виконання програми. Для оголошення константи у програмі використовується специфікатор `const`. При цьому зазначається тип константи і обов'язково надається значення, наприклад:

```
const double Pi = 3.14159;
```

У якості значення константи можна подавати константний вираз, який містить раніш оголошені константи та змінні. Наприклад:

```
const double Pi2 = 2 * Pi, k = Pi / 180;
```

4. Математичні функції

Математичні функції широко використовуються для записування різних математичних виразів. Список математичних функцій C++ наведено у табл. 2.2.

Таблиця 2.2

Основні математичні функції C++ та класу Math (.NET)

Функція C++	Опис	Клас Math
int abs (int i)	модуль (абсолютне значення) цілого числа $x - x $	Math:: Abs (x)
double fabs (double x)	модуль дійсного числа $x - x $	Math:: Abs (x)
double sqrt (double x)	корінь квадратний $-\sqrt{x}$	Math:: Sqrt (x)
double pow (double x, double y)	піднесення x до степеня $y - x^y$	Math:: Pow (x,y)
double exp (double x)	експонента e^x	Math:: Exp (x)
double log (double x)	натуральний логарифм $-\ln(x)$	Math:: Log (x)
double log10 (double x)	десятковий логарифм $-\lg(x)$	Math:: Log10 (x)
–	логарифм x за основою $N - \log_N(x)$	Math:: Log (x,N)
double cos (double x)	косинус $-\cos(x)$	Math:: Cos (x)
double sin (double x)	синус $-\sin(x)$	Math:: Sin (x)
double tan (double x)	тангенс $-\operatorname{tg}(x)$	Math:: Tan (x)
double acos (double x)	арккосинус $-\arccos(x)$	Math:: Acos (x)
double asin (double x)	арксинус $-\arcsin(x)$	Math:: Asin (x)
double atan (double x)	арктангенс $-\operatorname{arctg}(x)$	Math:: Atan (x)
double cosh (double x)	гіперболічний косинус $-(e^x + e^{-x})/2$	Math:: Cosh (x)
double sinh (double x)	гіперболічний синус $-(e^x - e^{-x})/2$	Math:: Sinh (x)
double ceil (double x)	округлення доверху: найменше ціле, не менше за x	Math:: Ceiling (x)
double floor (double x)	округлення донизу: найбільше ціле, не більше за x	Math:: Floor (x)

Стандартні математичні функції C++ розміщені у заголовному файлі (бібліотеці) **math.h**, який долучають директивою **#include <math.h>**.

До речі, константи **math** (наприклад, **M_PI** ($\pi \approx 3.14159$), **M_E** (число Ейлера $e \approx 2.71828$, яке є основою натуральних логарифмів) та багато інших) не вказані в стандартному C/C++. Щоб використовувати їх, слід спочатку визначити **_USE_MATH_DEFINES**, а тоді долучити **cmath** або **math.h**.

```
#define _USE_MATH_DEFINES
```

```
#include <math.h>    // для C або #include <cmath> для C++
```

У платформі .NET існують свої відповідні константи **Math::PI** та **Math::E**. Крім того, в ній є досить зручна функція **Math::Log(x,N)**, яка дозволяє обчислити значення логарифма за будь-якою основою N (для обчислення такого логарифма в "чистому" C доведеться використовувати вираз $\log(x)/\log(N)$).

5. Правила записування арифметичних виразів

Вираз, який завершується крапкою з комою, є **оператором**. Оператор задає закінчений опис певної дії.

Арифметичний вираз складається із операндів, арифметичних операцій (+, -, *, /, %, ++, --) і оператора присвоювання (=).

Таблиця 2.3

Арифметичні операції

Позначення	Операція	Типи операндів і результату	Приклади
+	додавання	арифметичний, вказівник	$x + y$
-	віднімання та унарний мінус	арифметичний, вказівник	$x - y$
*	добуток	арифметичний	$x * y$
/	ділення	арифметичний	x / y
%	остача від ділення цілих чисел	цілий	$i \% 6$
++	збільшення на одиницю (інкремент)	арифметичний, вказівник	$i++$; $++i$
--	зменшення на одиницю (декремент)	арифметичний, вказівник	$i--$; $--i$

Результатом операції обчислення остачі від цілочисельного ділення (%) є залишок від ділення першого операнда на другий. Операндами цієї операції мають бути цілі числа.

```
int n = 49, m = 10, x, y;
x = n / m;    // x = 4
y = n % m;    /* y = 9 */
```

Текст, який розміщений після двох скісних рисок (//) (або у середині /* */), є **коментарем** і не береться до уваги при компілюванні файлу.

Операції ++ (**інкремент**) та -- (**декремент**) є унарними, тобто мають лише один операнд. Операція ++ додає одиницю до операнда, операція -- віднімає одиницю від операнда. Ці операції можуть бути записані як праворуч, так і ліворуч операнда. Залежно від місця розміщення операції відносно операнда розрізняють дві форми цих операцій: префіксну та постфіксну. У префіксній формі, в якій операцію розміщують перед операндом, наприклад: ++i, --j, спочатку збільшується або зменшується на одиницю значення змінної, а вже потім ця змінна з її новим значенням бере участь в арифметичному виразі. У постфіксній формі цих операцій, навпаки, операцію розміщують після операнда, наприклад: i++, j--, і у виразі спочатку використовується поточне значення цієї змінної, а потім збільшується або зменшується її значення.

Чотири нижченаведені оператори дають однакові результати, але мають різницю при використанні у виразах:

```
int j, i = 1;
i = i + 1;  i += 1;  ++i;  i++;
```

Наведемо ще три приклади використання операцій інкремента. Всі вони будуть виконуватись за початкового значення $i = 1$:

1) $j = ++i * ++i$; // 1) $i=i+1=2$, 2) $i=i+1=3$, 3) $j=i*i=3*3=9$. Результат $i=3$, $j=9$.

- 2) $j = i++ * i++;$ // 1) $j=i*i=1*1=9$, 2) $i=i+1=2$, 3) $i=i+1=3$. Результат $i=3$, $j=1$.
3) $j = i++ * ++i;$ // 1) $i=i+1=2$, 2) $j=i*i=2*2=4$, 3) $i=i+1=3$. Результат $i=3$, $j=4$.

Обчислення в арифметичних виразах виконуються зліва направо згідно з таким **пріоритетом** операцій:

- 1) стандартні функції, ++, --;
- 2) множення (*), ділення (/), остача від ділення (%);
- 3) додавання (+) та віднімання (-).

Вирази у круглих дужках виконуються першочергово.

Для здобуття правильного результату слід дотримуватися таких правил записування арифметичних виразів в операторах C++:

- кожна команда (інструкція) має завершуватись крапкою з комою (;);
- мова C++ є чутлива до регістру, тобто x та X – це дві різні змінні;
- аргумент функції завжди записують у круглих дужках;
- знаки множення не можна пропускати ($3ab \rightarrow 3*a*b$);
- якщо знаменник або чисельник має операції (+, -, *, /), то його слід записувати у круглих дужках;
 - для записування раціональних дробів, у чисельнику або знаменнику яких є числові константи, хоча б одну з цих констант слід записати як дійсне число із зазначенням десяткової крапки, наприклад, $\frac{2}{k}$ записують як $2.0/k$;
 - радикали (тобто корінь кубічний і вище) замінюють на дробові степені, наприклад, $\sqrt[3]{x+1}$ записують як $\text{pow}(x+1, 1/3.0)$;
 - слід враховувати правила зведення типів, оскільки в арифметичних виразах можуть брати участь різнотипні дані та відбувається зведення типів.

6. Оператори присвоювання

Оператор присвоювання – основний оператор програмування – має формат:

$\langle \text{ім'я_змінної} \rangle = \langle \text{вираз} \rangle;$

Цей оператор обчислює значення *виразу* і надає здобуте значення *змінній*; при цьому слід враховувати відповідність типів виразу і змінної.

У мові C++, крім простого присвоювання, є і складені операції присвоювання: $+=$ (присвоювання з додаванням), $- =$ (присвоювання з відніманням), $* =$ (присвоювання з множенням), $/ =$ (присвоювання з діленням) тощо. Наприклад, вираз $x += y$ є еквівалентний до виразу $x = x + y$, але записується компактніше і виконується швидше.

Особливістю оператора присвоювання в C+ є те, що він може використовуватись у виразах і допускає багатократне застосування, наприклад:

$a = b = c = x * y;$

Виконується ця команда справа наліво, тобто спочатку обчислюється значення виразу $x * y$, після чого це значення присвоюється c , потім b і лише потім a .

7. Зведення типів

Перетворення типів виконуються, якщо операнди, які входять до виразу, мають різні типи. Зведення типів здійснюється автоматично за правилом: **менш**

точний тип зводиться до більш точного, тобто до типу того операнда, який має більший розмір. Наприклад, якщо в арифметичному виразі беруть участь коротке ціле `short` і ціле `int`, результат зводиться до `int`, якщо ціле і дійсне – до дійсного.

Розглянемо ще кілька правил зведення типів.

1) Результат операції ділення буде цілим числом, якщо ділене і дільник є цілими, і дійсним числом, якщо один з операндів є дійсного типу. Наприклад, результатом $2/3$ буде `0`, а результатом $2.0/3$ (або $2./3$) – `0.666(6)`.

2) При присвоюванні результат зводиться до типу змінної, яка стоїть ліворуч “=”; при цьому тип може як підвищуватися, так і знижуватись, наприклад:

```
float a = 5.1, b = 1.5;
int c = a * b; // c=7, а дробова частина від результату 7.65 буде втрачена
```

2) Зведення типів може бути явним із зазначенням конкретного типу:

(`<тип>`) `<арифметичний_вираз>`

Наприклад:

```
int m = 2, n = 3;
double a = (double) m/n; // в результаті a=0.66(6)
```

Іншим рішенням може бути домноження чисельника чи знаменника на `1.0`:

```
double a = (m*1.0)/n; // в результаті a=0.66(6)
```

Приклади програм з лінійною структурою в C++


Приклад 1. Розробити схему алгоритму і створити програмний проект у

Visual C++ для обчислення $y = \frac{0.2x^2 - x}{(\sqrt{3} + x)(1 + 2x)} + \frac{2(x-1)^3}{\sin^2 x + 1}$, де x – довільна змінна, яку слід ввести.

Розв’язок. Послідовність виконання завдання:

1) Запустити Visual Studio.

2) Виконати команду *Создать проект / Visual C++ / CLR / Консольное приложение CLR*. Не закриваючи вікно *Создать проект*, натиснути кнопку *Обзор*, вибрати теку для зберігання та задати ім’я програмного проекту. Після цього натиснути кнопку *ОК* та приступити до написання програмного коду.

3) Перед запуском на виконання програмний проект слід зберегти командою *Файл / Сохранить все* (чи то через ярлик , чи клавішами *Ctrl+Shift+S*).

4) Для запуску проекту слід виконати команду меню *Отладка / Начать отладку* чи то натиснути на клавіатурі кнопку *F5*.

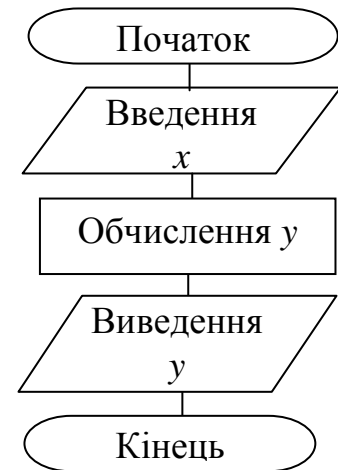
Текст програмного коду та схема алгоритму:

```
#include <iostream>
#include <math.h>
using namespace std;
```

```
int main ()
{
    setlocale(0, ".1251");
    double y, x;
    cout<<"Введіть значення x= ";
    cin>>x;
    y=(0.2*x*x-x)/((sqrt(3.)+x)*(1+2*x)) +
        2*pow(x-1,3)/(pow(sin(x),2)+1);
    cout<<"Результат y= "<< y <<endl;
    system ("pause>>void");
    return 0;
}
```

Результати роботи консольного додатка:

```
Введіть значення x= 2.15
Результат y= 1.72928
```



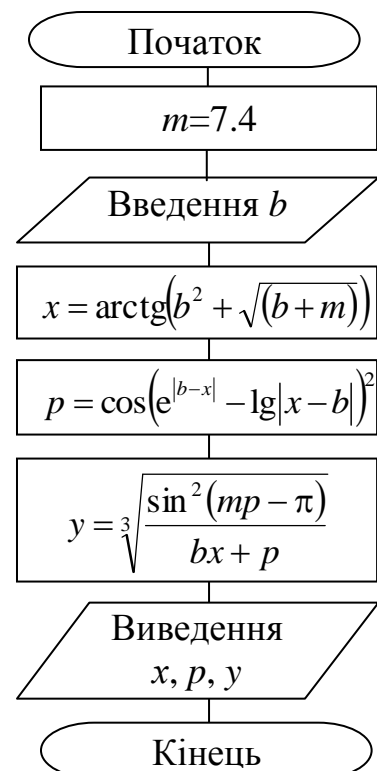
Приклад 2. Розробити схему алгоритму і програмний проект для обчислення $y = \sqrt[3]{\frac{\sin^2(mp - \pi)}{bx + p}}$; $x = \arctg(b^2 + \sqrt{(b+m)})$ і $p = \cos(e^{|b-x|} - \lg|x-b|)^2$. Значення $m = 7.4$ задати як константу, а довільне значення змінної b слід ввести.

Текст програми та схема алгоритму:

```
#include <iostream>
#include <math.h>
using namespace std;
int main ()
{
    setlocale(0, ".1251");
    const double m=7.4, pi=3.1415;
    double b, x, p, y;
    cout<<"Введіть значення b= "; cin>>b;
    x=atan(b*b+sqrt(b+m));
    p=cos(pow(exp(fabs(b-x))-
        log10(fabs(x-b)),2));
    y=pow(pow(sin(m*p-pi),2)/(b*x+p),1./3);
    cout<<"Результати:\n x= "<< x <<endl;
    cout<<" p= "<< p <<endl;
    cout<<" y= "<< y <<endl;
    system ("pause>>void");
    return 0;
}
```

Результати роботи консольного додатка:

```
Введіть значення b= 8.4
Результати:
x= 1.55738
p= -0.128333
y= 0.3709956
```



Приклад 3. Обчислити ємність конденсатора за формулою $C = \frac{\varepsilon_0 \cdot \varepsilon \cdot S}{d}$,

де $\varepsilon_0 = 8.85 \cdot 10^{-12}$ Ф/м, $\varepsilon = 2.8$ Ф/м, S – площа кожної пластини конденсатора, d – відстань між пластинами, значення яких Ввести з клавіатури.

Текст програмного коду:

```
#include <iostream>
#include <math.h>
using namespace std;
int main ()
{
    setlocale(0, ".1251");
    double d, S, C;
    const double E0 = 8.85e-12, E = 2.8;
    cout<< "Введіть значення відстані між пластинами d= " ;
    cin>>d;
    cout<< "Введіть значення площі пластини конденсатора S= ";
    cin>>S;
    C=(E0*E*S)/d;
    cout<<"Ємність конденсатора C= "<< C <<endl;
    system ("pause>>void");
}
```

Результати роботи консольного додатка:

```
Введіть значення відстані між пластинами d= 0.0004
Введіть значення площі пластини конденсатора S= 50000
Ємність конденсатора C= 0.0030975
```

Приклад 4 Створити проект для обчислення опору електричного кола, яке складається з двох поєднаних опорів, у два варіанти: 1) за паралельного з'єднання опорів та 2) за послідовного з'єднання опорів в електричному колі.

Розв'язок. Нагадаємо, що формула для обчислення опору електричного кола за паралельного з'єднання опорів має вигляд $R = \frac{r1 \cdot r2}{r1 + r2}$, а за послідовного з'єднання опорів – $R = r1 + r2$.

Обчислення для двох випадків сформуємо в програмі у двох окремих командних кнопках на формі:

Для створення програмного проекту з формою слід запустити Visual Studio та виконати команду *Создать проект / Visual C++ / CLR / Приложение Windows Forms*. Не закриваючи вікно *Создать проект*, натиснути кнопку *Обзор*, вибрати теку для зберігання та задати ім'я програмного проекту. Натиснути кнопку *OK* та приступити до створення форми і написання програмного коду.

Текст програми:

```
// Обчислити для паралельного з'єднання
private: System::Void button1_Click(System::Object^sender, System::EventArgs^e)
{
    double r1, r2, R;
    r1 = Convert::ToDouble(textBox1->Text); // значення 1-го опору r1
    r2 = Convert::ToDouble(textBox2->Text); // значення 2-го опору r2
    R = r1*r2/(r1+r2);
    textBox3->Text = Convert::ToString(R);
}

// Обчислити для послідовного з'єднання
private: System::Void button2_Click(System::Object^sender, System::EventArgs^e)
{
    double r1, r2, R;
    r1=Convert::ToDouble(textBox1->Text); // значення 1-го опору r1
    r2=Convert::ToDouble(textBox2->Text); // значення 2-го опору r2
    R=r1+r2;
    textBox3->Text=Convert::ToString(R);
}

// Очищення
private: System::Void button3_Click(System::Object^sender, System::EventArgs^e)
{
    textBox1->Clear(); textBox2->Clear(); textBox3->Clear();
}

// Вихід
private: System::Void button4_Click(System::Object^sender, System::EventArgs^e)
{
    Close();
}
```

Питання та завдання для самоконтролю

- 1) Який процес називають лінійним?
- 2) Які стандартні типи даних C++ Вам відомі?
- 3) Яких значень набудуть змінні j та i після обчислення для `int j, i=2;`
 - а) $j = i++ + i++;$
 - б) $j = ++i + ++i;$
 - в) $j = i++ + ++i;$
 - г) $j = ++i + i--;$
- 4) Запишіть константу $0.2731e3$ у фіксованому форматі.
- 5) Запишіть число 0.0001 в експоненційній формі з рухомою крапкою.
- 6) Запишіть вираз $y = x^3 + \sin 2x$ засобами C++.
- 7) Якого значення набуде змінна y після обчислення:
 - а) $y = (1 + 2) / (3 + 2);$
 - б) $y = 1 + 2 / 3 + 2;$
 - в) $y = 1 + 2.0 / 3 + 2;$
 - г) $y = (1 + 2) / (3.0 + 2).$

Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
- 2) У протоколі лабораторної роботи записати мовою С++ арифметичні вирази, наведені в табл. 2.4 ... 2.6 відповідно до індивідуального варіанта.
- 3) У протоколі лабораторної роботи скласти схеми алгоритмів і написати програми мовою С++ для розв'язання завдань, поданих в табл. 2.7 ... 2.9 відповідно до індивідуального варіанта.
- 4) Створити на комп'ютері програмні проекти в середовищі Visual С++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Варіанти індивідуальних завдань з теми “Лінійні обчислювальні процеси”

Таблиця 2.4

Варіанти арифметичних виразів

№ вар.	Арифметичний вираз	№ вар.	Арифметичний вираз	№ вар.	Арифметичний вираз
1	$Z = \frac{2t + y \cos t}{\sqrt{y + 4.831}}$	2	$D = y^2 + \frac{0.5n + 4.8}{\sin y}$	3	$Q = \frac{\sqrt{k + 2.6p \sin k}}{x - d^3}$
4	$F = \ln(d) + \frac{3.5d^2 + 1}{\cos(2y + 2.3)}$	5	$R = \frac{\sin(2t + 1)^2 + 0.3}{\ln(t + y)}$	6	$L = \cos^2 c + \frac{3t^2 + 4}{\sqrt{c + t}}$
7	$U = \frac{\ln(k - y) + y^4}{e^y + 2.355k^2}$	8	$A = \frac{\sin(2y + h) + h^2}{e^h + y}$	9	$R = \frac{\sin^2 y + 0.3d}{e^y + \ln(d)}$
10	$G = \frac{9.33w^3 + \sqrt{w}}{\ln(y + 3.5) + \sqrt{y}}$	11	$P = \frac{e^{y+2.5} + 7.1h^3}{\ln \sqrt{y} + 0.04h}$	12	$U = \frac{\ln(2k + 4.3)}{e^{k+y} + \sqrt{y}}$
13	$D = \frac{7.8a^2 + 3.52t}{\ln(a + 2y) + e^y}$	14	$F = \frac{2 \sin(0.354y + 1)}{\ln(y + 2j)}$	15	$T = \frac{\sin(2 + u)}{\ln(2y + u)}$
16	$L = \frac{0.81 \cos i}{\ln(y) + 2i^3}$	17	$W = \frac{4t^3 + \ln(r)}{e^{y+r} + 7.2 \sin r}$	18	$G = \frac{e^{2y} + \sin(f + 3)}{\ln(3.8y + f)}$
19	$N = \frac{m^2 + 2.8m + 0.355}{\cos 2y + 3.6}$	20	$H = \frac{y^2 - 0.8y + \sqrt{y}}{23.1n^2 + \cos n}$	21	$Z = \frac{\sin(p + 0.4)^2}{y^2 + 7.325p}$
22	$T = \frac{2.37 \sin(t + 1)}{\sqrt{4y^2 - 0.1y + 5}}$	23	$R = \frac{\sqrt{\sin^2 y + 6.835}}{\ln(y + k) + 3y^2}$	24	$W = \frac{0.004v + e^{2y}}{e^{y/2}}$
25	$V = \frac{(y + 2w)^3}{\ln(y + 0.75)}$	26	$E = \frac{\ln(0.7y + 2q)}{\sqrt{3y^2 + 0.5y + 4}}$	27	$T = \frac{0.355h^2 - 4.355}{e^{y+h} + \sqrt{2.7y}}$
28	$S = \frac{4.351y^3 + 2t \ln(t + x)}{\sqrt{\cos 2y + 4.351}}$	29	$K = \frac{2t^2 + 3l + 7.2}{\ln(y) + e^{2l}}$	30	$N = \frac{3y^2 + \sqrt{y + 1}}{\ln(p + y) + e^p}$

Таблиця 2.5

Варіанти арифметичних виразів

№ вар.	Арифметичний вираз	№ вар.	Арифметичний вираз
1	$F = \cos(x^2 + 2) + \frac{3.5x^2 + 1}{\cos^2 y}$	2	$T = \frac{\sqrt{x+b-a} + \ln(x)}{\operatorname{ctg}(b+a)}$
3	$R = \frac{\sin(x^2 + a)^3 + 4.3^a}{\cos^3 x^4}$	4	$D = \frac{K^{-arx} - a\sqrt{6} - \cos(3ab)}{\sin^2(a \arcsin x + \ln y)}$
5	$L = \operatorname{ctg}^2 c + \frac{2x^2 + 5}{\sqrt{c+t}}$	6	$U = \frac{\ln(x^3 + y) - y^4}{e^y + 5.4k^3}$
7	$P = \frac{a^5 + \arccos(a + x^3) - \sin^4(y - c)}{\sin^3(x + y) + x - y }$	8	$A = \frac{\operatorname{tg}(y^3 - h^4) + h^2}{\sin^3 h + y}$
9	$F = \frac{\sqrt{(2+y)^2} + \sqrt[7]{\sin(y+5)}}{\ln(x+1) - y^3}$	10	$R = \frac{\cos^2 y + 2.4d}{e^y + \ln(\sin^2 x + 6)}$
11	$G = \frac{\operatorname{tg}(x^4 - 6) - \cos^3(z + xy)}{\cos^4 x^3 c^2}$	12	$F = \frac{\sqrt{ x + \cos^3 x + z^4}}{\ln x - \arcsin(bx - a)}$
13	$U = \frac{\operatorname{tg}^3 y + \sin^5 x \sqrt{b-c}}{\sqrt{a-b+c}}$	14	$D = \frac{\cos(x^3 + 6) - \sin(y - a)}{\ln x^4 - 2 \sin^5 x}$
15	$P = \frac{a^5 + \sin^4(y - c)}{\sin^3(x + y) + x - y }$	16	$G = \frac{\operatorname{tg}(x^4 - 6) - \cos^{3x}(z + x^3 y)}{\cos^2 x^3 + c^2}$
17	$R = \frac{\cos^3 y + 2^x d}{e^y + \ln(\sin^2 x + 7.4)}$	18	$S = \frac{4.351y^3 + 2t \ln(t)}{\sqrt{\cos 2y + 1}}$
19	$U = \frac{e^{x^3} + (\cos^2 x - 4)}{\operatorname{arctg} x + 5.2y}$	20	$N = \frac{\sqrt[5]{z + \sqrt{zx}}}{e^x + a^5 \operatorname{arctg} x}$
21	$I = \frac{2.33 \ln \sqrt{1 + \cos^2 y}}{e^y + \sin^2 x}$	22	$K = \frac{\sqrt{(3a+x)^6 - \ln x}}{e^{a+x} + \arcsin 6x^2}$
23	$K = \frac{\cos^3 y+x - (x+y)}{\operatorname{arctg}^4(x+a)x^5}$	24	$R = \frac{\sqrt{\sin^2 y + 6.835}}{\ln(y+k) + 3y^2}$
25	$R = \frac{a}{x-a} + \frac{b^x + \cos^3 x}{\lg^3 a + 4.5}$	26	$G = \frac{9.33w^3 + \sqrt{w}}{\ln(y+3.5) + \sqrt{y}}$
27	$L = \frac{\sqrt{e^x - \cos^4(x^2 a^5)} + \operatorname{arctg}^4(a - x^5)}{\sqrt{ a+xc }}$	28	$f = \frac{\cos^7 bx^5 - (\sin a^2 + \cos(x^3 + z^5))}{(\arcsin a^2 + \arccos(x^7 - a^2))}$
29	$N = \frac{m^2 + 2.8m + 0.355}{\cos 2y + 3.6}$	30	$H = \frac{y^2 - 0.8y + \sqrt{y}}{23.7n^2 + \cos n}$

Варіанти арифметичних виразів

№ вар.	Арифметичний вираз
1	$V1 = \sqrt{x^2 + \left(\sqrt{\arctg x - e^2} / \sin^2(x^3 + 1.8)\right)^4} + 2.8^{\sqrt{x}}$
2	$V2 = (\sin x - 5.4)^{3x} + \sqrt[3]{\lg(x - 1.5)^2} + x^{3.5}$
3	$V3 = x^{2.8} / \left(\cos^2(x^3 - 1.5)^4 + \sqrt{ x }\right) - \arctg(x / \ln x)^5$
4	$V4 = \sin^5\left(x^4 - \sqrt[3]{\lg^4(x^2 - \ln^2(x - 1.8))}\right) + \arctg^2 x$
5	$V5 = (15.4^x - x^{3.9}) / \sqrt{x^2 + \lg^2 \ln^3 x^3 - 1.8 } + 9^{5.3}$
6	$V6 = e^{\sqrt{\lg^3(x^2 - 1.8)^5}} + x^{4.5} / \arctg(x^2 + a^2)^4 - \sqrt{x^{3.2}}$
7	$V7 = (\cos^3 x^{1.5} + \sin^2 x^3)^4 / \left(\lg^2(x + e^{\sqrt[3]{x+1.8}}) + \sqrt{x}\right)$
8	$V8 = \lg^4\left(\ln^3(x^2 + \sqrt{ x }) / (x^3 + e^x)^3\right)^5 - x^{3.5} / \sin^2(x^3 + 1.8)$
9	$V9 = \cos^5\left(x^2 + \arctg\sqrt{ x - 1.8 }\right) / \sin^2(x^2 + 1.5)^5 + \sqrt[3]{x^{3.5}}$
10	$V10 = \arctg^5\left(\sin^3(x^2 + 1.8)^5 - \sqrt{x}\right)^4 - e^{3.8} / (x^{4.5} + \sqrt{ x })$
11	$V11 = \cos^3\left(x^4 + \lg^2 \ln^3(\sqrt{x} - \sqrt[3]{ x })^2\right) + (4.8^{\sqrt[3]{x}} - \sqrt{ x })^5$
12	$V12 = \sin^8\left(\sqrt{x} + \sqrt[3]{x^2 + 1.8} / \cos^2(x^3 - 1.5)^6\right) - x^{3.7}$
13	$V13 = e^x \left(\cos^2(x - \sqrt[3]{x^4 + 5.3})^4 + \arctg^3 x^2\right)^5 + x^2 / (1 + x^{6.6})$
14	$V14 = \sqrt[3]{ x + \sqrt{x^3 + 1.3}} / \cos^2(x^3(1 + x)^4) / e^{\sqrt{x}} - x^{7.5}$
15	$V15 = \sin^5\left(\sqrt{ x } + \cos^3(x^2 + 5.4)\right) - \arctg(e^{\sqrt{x}} + 5.8^{3.7})$
16	$V16 = \left(x^2 + \sqrt{\sin^2 x - \ln(x^2 - 3)}\right) / \lg^2\left(x + \sqrt[3]{5.5x + \ln x}\right) - e^{2.5+x^2}$
17	$V17 = \cos^2(x^2 + \sqrt{x+2}) / \sin(x^2 + \sqrt{x}) + \ln^2 x / (\lg x + e^{\sqrt{x}})$
18	$V18 = \sqrt{x^2 + \sin(\sqrt{x} + 2x)} - e^{2x+\sqrt{x}} / (\cos^2 x + \lg^2(\ln x))$
19	$V19 = \sqrt[3]{x + \cos^2 x + \sin x^2 + \lg x} / (x^2 + \ln^2 x^3 - e^{\sqrt{x}})$
20	$V20 = \sqrt{x + \sqrt{x + \sqrt{x + \sin^2 x}}} / \cos(x^2 + \ln^2(1 + e^{\sqrt{x}}))$
21	$V21 = \left(x^2 + \cos\left(x + \ln \sqrt{x^3 + 1.8}\right)^2 + \sqrt{ x }\right) / \lg x + e^{\sqrt{x}} $
22	$V22 = \sin^3\left(x^{2.3} \sqrt{x + \sqrt{x^2 + 1.5}}\right) / \arctg^2(x^2 + 2.5e^x)$
23	$V23 = \cos^2\left(x + \sin\left(\sqrt{x^3 + \sqrt{x+1.5}} - \ln^2 x\right) + e^x\right)$

Закінчення табл. 2.6

№ вар.	Арифметичний вираз
24	$V24 = (x^{4.5} e^{\sqrt{x}}) / \ln^2(x^2 + \cos(x + \sqrt[3]{x})) - \sin^3 x - e^3 $
25	$V25 = \arctg^2(x^2 + \cos(\sqrt{x + x^2 - e^x }) / \lg^2(x^3 - \sin x))$
26	$V26 = \sin^2(x^2 + \sqrt[3]{x + \cos^2 x }) / \ln^2(\sqrt{x} - \arctg^2 x)^2$
27	$V27 = \ln^2(\sqrt{ x } + x^2 + \sin x) / \lg^2(e^{\sqrt{x}} + x^4)$
28	$V28 = \lg(\ln^2(x) x^2 + \sqrt{x^2 + 1.5}) / e^{\sqrt{x}} + \cos^3 x^2$
29	$V29 = \cos^3(\arctg^2(x + \sqrt[3]{x}) + \sin^2 x) / \sin \lg^2(x + 2.8) $
30	$V30 = \lg^5(x^3 \ln^6(x^2 + 1.7) + e^{\sqrt{x}}) - \arctg \sqrt[3]{x} + \tg^3 x^8 $

Таблиця 2.7

Варіанти завдань для програм з лінійною структурою

(довільне значення x слід Ввести)

№ вар.	Функція	№ вар.	Функція
1	$y = \frac{2x^2 - \sin^2 x}{\cos(2x) + x^2} - \frac{x+1}{\ln x}$	2	$y = \frac{\ln x^2 + \cos^2 x}{\cos(2x) + x^2} + \frac{\sqrt[3]{x}}{x}$
3	$y = \frac{\ln x^2 + 2\cos^2 x}{\cos(2x)^2} + \frac{\sqrt[3]{x}}{x}$	4	$y = \frac{2\cos^2 x}{1 + x\cos(2x)} + \frac{0.3^x}{x\ln x - 2\sin^2 x}$
5	$y = \frac{x + 2x + \sin x}{\cos^2 x + x^2} + \frac{0.3^x}{\ln x}$	6	$y = \frac{2x + \sin x}{\cos^2 x + x^2} + \frac{0.5^x}{\sqrt{x}}$
7	$y = \frac{\sin x - x^2}{2x+1} + \frac{(1+x)^x}{1+3x}$	8	$y = \frac{x - \ln x}{2x-1} + \frac{2x-1}{x^2+3x}$
9	$y = \frac{\ln x + 2x}{x^2+1} + \frac{x+1}{2x^2+1}$	10	$y = \frac{3x^2+2x}{\sin x + x^2} - \frac{2x}{(1+x^2)(1+2x)}$
11	$y = \frac{4x^2+3x}{(1+x)(1+2x)} + \frac{2x+1}{\sin x+1}$	12	$y = \frac{(2x^2-1)}{x^2+\sin} - \frac{2x+1}{(x+2)(x+3)}$
13	$y = \frac{(4x^2-2)(x+2)}{2x+3} + \frac{x^2 \sin x}{2x+1}$	14	$y = \frac{x^2+2\sin x}{2x+1} + \frac{\sqrt{x} - \cos x}{(2x+1)(\ln x^2+1)}$
15	$y = \frac{x^2+2(x-1)}{(x+1)(x+\sqrt{3})} + \frac{2\sin^2 x}{2x+3^x}$	16	$y = \frac{4x^2-3^x}{2x^2+1} + \frac{\ln x}{2x+3}$
17	$y = \frac{3x-2}{(2x+3)(x+1)} + \frac{\sin 2x}{(x^2+1)(x+2)}$	18	$y = \frac{x^2-2x}{(2x+3)(x+4)} + \frac{\sqrt[3]{x}}{2x+3}$
19	$y = \frac{x^2+1}{x^3+3} + \frac{\sin x}{2x+3}$	20	$y = \frac{4x^2+3x}{3x+4} + \frac{\sin x}{2\cos x+1}$

Закінчення табл. 2.7

№ вар.	Функція	№ вар.	Функція
21	$y = \frac{3x+2}{2x+3} + \frac{x^2}{(2x+1)(\sin x+2)}$	22	$y = \frac{4x - \sin x}{x^2 + 3x + 1} + \frac{3x^2 + 2^x}{(x+1)(x^2+1)}$
23	$y = \frac{2x + \sin x}{(x+2)(x+\sqrt{x})} + \frac{4x}{(2x+\sqrt[3]{3})(x^2+1)}$	24	$y = \frac{3x+4}{(x+3)(x+1)^2} + \frac{2x-1}{(x+\sin x)(\ln x+1)}$
25	$y = \frac{4x}{(x+\sin)^2} + \frac{2x+\sqrt[3]{x}}{(x^2+1)(x+1)}$	26	$y = \frac{2x+3^x}{(x+1)(x+3)} + \frac{x^2+\sqrt{x}}{(2x+1)(x+\sin x)}$
27	$y = \frac{2x+1}{x+1} + \frac{x^2+\sqrt{x}}{(2x+1)(x+\sin x)}$	28	$y = \frac{3x^2-\sqrt[3]{x}}{(2x+1)(1+x)} + \frac{2x+1}{(x+3)}$
29	$y = \frac{2x^2+1}{x+\sin(x+1)} + \frac{x-3^x}{(2x+1)(x+2)}$	30	$y = \frac{x^2+\sin 2x}{2\sqrt{x}+3x} + \frac{x^2+1}{(x+2)(x+3)}$

Таблиця 2.8

Варіанти завдань з лінійною структурою

(перше зі значень параметрів задати як константу, друге – ввести з екрана)

№ вар.	Функція $y = f(x)$	Значення параметрів
1	$y = a \sin^2 b + b \cos^2 a; a = \sqrt[3]{b+c}; b = \sqrt{x}$	$x = 1.52; c = 5$
2	$y = a^2 + b^2; a = \ln x ; b = e^k + a$	$x = 5.3; k = 3$
3	$y = e^x + 5.8^c; c = a^2 + \sqrt{b}; a = b^3 + \ln b $	$x = 2.5; b = 7$
4	$y = \sqrt[3]{a-b}; a = \lg x; b = \sqrt{x^2+t^2}$	$x = 1.7; t = 3$
5	$y = a^3 / b^2; a = e^{\sqrt{ x }}; b = (\sin p^2 + x^3)$	$x = 2.1; p = 2$
6	$y = p^2 + t^4; p = x^2 - \sqrt{ x }; t = \sqrt[3]{x+a^2}$	$x = 4; a = 3.7$
7	$y = c^3 / \cos c; c = a^2 + b^2; a = \sqrt{ x } + e^{\sqrt{b}}$	$x = -11; b = 12.5$
8	$y = \sin^3(a+b); a = t^3 + \sqrt{b}; b = \lg^2 x $	$x = 10.9; t = 2$
9	$y = \arctg^3 x^2; x = p+k; k = \sqrt{p+t^2}$	$t = 4.1; p = 3$
10	$y = \cos^2(a+\sin b); a = \sqrt{ x }; b = x^4 + m^2$	$m = 2; x = 1.1$
11	$y = \sin^3 a + \cos^2 x; a = c+k^2; c = \arctg x $	$k = 7.2; x = 5$
12	$y = e^{\sqrt{ x }} + \cos x; x = a+c^3; a = \sin^5 b$	$b = 3; c = 1.7$
13	$y = a \cos x - b \sin x; x = \sqrt[3]{a-b}; a = t^2 b$	$t = 2.2; b = 3$
14	$y = \sqrt{x} \sin a + \sqrt{b} \cos x; a = \lg x ; b = x + p^3$	$x = 11; p = 2.6$

Закінчення табл. 2.8

№ вар.	Функція $y = f(x)$	Значення параметрів
15	$y = \lg a / \lg b; a = \sqrt{x^2 + b^2}; x = e^b + n$	$n = 9.1; b = 3$
16	$y = \ln x+t ; x = t^2 + p; t = \sqrt{m}$	$m = 3.8; p = 2$
17	$y = e^{a+b}; a = \lg t+b^2 ; t = b^2 + \sqrt{bx}$	$b = 3; x = 5.2$
18	$y = \sqrt[3]{x^2 + c^2}; x = e^{mk}; c = \cos^2 m + k^2$	$k = 2; m = 1.8$
19	$y = e^x + 5.8^c; c = a^2 + \sqrt{b}; a = b^3 + \ln b $	$x = 2.8; b = 3$
20	$y = x^3 / t^2; x = e^{\sqrt{p+a}}; t = p^3 + a^3$	$a = 2; p = 2.6$
21	$y = c^2 + \sqrt{ a }; c = \lg b ; a = (b+x)^3$	$b = 7; x = 2$
22	$y = \operatorname{arctg}^2 x ; x = t^3 + b^2; t = b^3 + e^{\sqrt{q}}$	$q = 2; b = 1.8$
23	$y = v^3 + \cos^2 w; v = \cos^2 a; w = \sqrt{a+ x }$	$x = 2.9; a = -0.9$
24	$y = x^2 + \sqrt[3]{ x }; x = \cos^2 b + \sin^2 a; a = \sqrt{b+t^2}$	$b = 7.1; t = 2$
25	$y = \sin^3 x + \cos x^2; x = \lg ct ; c = t^2 + \sqrt{a}$	$t = -3; a = 8.8$
26	$y = \lg^2 x+a ; x = \sqrt{a+b}; a = e^{t+b}$	$t = 2; b = 1.8$
27	$y = \operatorname{arctg}^3 p ; p = \sqrt{x^2 + a^2}; x = \sqrt{a} + \sqrt{b}$	$a = 7; b = 2.3$
28	$y = \sin^4(a^2 + b^2); a = \sqrt{b+t}; t = b^2 + k^3$	$b = 5; k = 2.8$
29	$y = \cos^3 x + a ; x = e^b; b = a + \sqrt{a+p^2}$	$a = -4; p = 3$
30	$y = \sin^4(a^2 + b^2); a = \sqrt{b+t}; t = b^2 + k^3$	$b = 2; k = 1.8$

Таблиця 2.9

Варіанти задач з лінійною структурою

№ вар.	Завдання
1	Трикутник задано координатами своїх вершин. Обчислити його площу, використовуючи формулу Герона: $S = p(p-a)(p-b)(p-c)$, де $p = (a+b+c)/2$; a, b і c – довжини сторін трикутника. Координати вершин ввести з клавіатури. Для обчислення довжини відрізка між точками $(x_1, y_1), (x_2, y_2)$ використовувати формулу $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
2	Знайти період T і частоту ν коливань у контурі, ємність конденсатора в якому – C , індуктивність – L . Значення C і L ввести з клавіатури. $T = 2\pi\sqrt{LC}, \nu = 1/T$
3	Обчислити довжину і площу кола для заданого радіуса. Значення радіуса ввести з екрана
4	Обчислити площу трикутника за трьома сторонами – a, b, c , – використовуючи формулу Герона (див. варіант 1). Довжини сторін ввести з клавіатури

№ вар.	Завдання
5	Знайти косинус кута поміж векторами $\vec{a} = (a_1, a_2)$ та $\vec{b} = (b_1, b_2)$ за формулою $\cos \alpha = (\vec{a} \cdot \vec{b}) / (\vec{a} \cdot \vec{b})$. Модуль вектора $ \vec{a} = \sqrt{a_1^2 + a_2^2}$. Скалярний добуток векторів обчислити за формулою $\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2$
6	Обчислити відстань від точки M до площин $22x - 4y - 20z - 45 = 0$ та $3x - y + 5z + 1 = 0$, використовуючи формулу відстані від точки до площини $\rho = \frac{ ax_0 + y_0 + cz_0 + d }{\sqrt{a^2 + b^2 + c^2}}$. Координати точки M ввести з екрана
7	Радіолокаційна станція випромінює $n = 1000$ імпульсів за 1 с з довжиною хвилі $\lambda = 0,03$ м. Потужність одного імпульсу $P_1 = 7 \cdot 10^{-4}$ Вт, а тривалість $\tau = 3 \cdot 10^{-7}$ Вт. Обчислити енергію одного імпульсу $W_1 = P_1 \tau$; середню корисну потужність станції $P = W_1 k$; кількість довжин хвиль в одному імпульсі $k = c \tau / \lambda$, якщо $c = 30$ ввести з екрана
8	Обчислити корені рівняння $ax^2 + bx + c = 0$, заданого коефіцієнтами a, b, c (припускаючи, що $a \neq 0$ і що корені є дійсні). Значення $a = 2, b = -8, c = -10$ ввести з екрана
9	Обчислити корінь рівняння $2x/a + b - 12 = 0$ за різних значень параметрів a та b . Значення a, b ввести з екрана
10	Обчислити гіпотенузу і площу прямокутного трикутника за двома заданими катетами. Довжини катетів ввести з екрана.
11	Тіло рухається за законом $S = t^3 - \sqrt{t}$. Обчислити швидкість тіла і відстань у момент часу t . Значення t ввести з екрана. (Функція швидкості є похідною від функції відстані)
12	Обчислити катет та площу прямокутного трикутника за заданими катетом і гіпотенузою. Довжини катета й гіпотенузи ввести з екрана
13	Обчислити $Z = (v_1 + v_2 + v_3)/3$, де v_1, v_2, v_3 – об'єми куль з радіусами R_1, R_2, R_3 відповідно. Об'єм кулі обчислити за формулою $V = 4/3 \pi R^3$. Значення радіусів ввести з екрана
14	У коливальному контурі ємність конденсатора $C = 10^{-6}$ Ф, індуктивність котушки $L = 0,04$ Гн, амплітуда напруги на конденсаторі $U = 100$ В. Обчислити амплітуду сили струму $I = U \sqrt{C/L}$, повну енергію $W = LI^2 / 2$
15	Чотирикутник задано координатами власних вершин. Обчислити його периметр. Вершини ввести з екрана
16	Обчислити значення функції $W = \text{sh}(x) \cdot \text{tg}(x+1) - \text{tg}^2(2 + \text{sh}(x-1))$, де $\text{sh}(x) = (e^x - e^{-x})/2$. Значення x ввести з екрана
17	При зміні сили струму в котушці, індуктивність якої $L = 0,5$ Гн, в $n = 2$ рази енергія магнітного поля змінилась на $\Delta W = 3$ Дж. Знайти початкові значення енергії $W_1 = \Delta W / (n^2 - 1)$ та сили струму $I_1 = \sqrt{2W_1 / L}$

Закінчення табл. 2.9

№ вар.	Завдання
18	Обчислити периметр трикутника, заданого координатами його вершин. Координати вершин ввести з екрана
19	Задано трикутник ABC довжинами власних сторін a, b, c , які слід ввести з екрана. Обчислити його бісектриси (бісектриса, проведена до сторони a , дорівнює $\sqrt{bc(a+b+c)(b+c-a)/(b+c)}$)
20	Задано трикутник ABC довжинами власних сторін a, b, c . Обчислити його медіани (медіана, проведена до сторони a , дорівнює $0,5\sqrt{2b^2 + 2c^2 - a^2}$. Значення a, b, c ввести з екрана
21	Обчислити $Z = (R_1 + R_2 + R_3)/3$, де R_1, R_2, R_3 – радіуси куль з об'ємами V_1, V_2, V_3 відповідно. Радіус кулі обчислити за формулою $R = \sqrt[3]{3V/4\pi}$. Значення об'ємів ввести з екрана
22	Задані довжини a, b і c сторін певного трикутника. Обчислити медіани трикутника, сторонами якого є медіани вихідного трикутника. Довжина медіани, проведеної до сторони a , дорівнює $0,5\sqrt{2b^2 + 2c^2 - a^2}$
23	За якого значення напруги на конденсаторі коливального контура (в долях амплітудного значення $u/U_{\text{макс}}$) і через який час (в долях періоду t/T) енергія електричного поля буде в n разів відрізняватися від енергії магнітного поля? Значення n ввести з екрана. $u/U_{\text{макс}} = \sqrt{n/(n+1)}$; $t/T = \arccos \sqrt{n/(n+1)}/(2\pi)$
24	Обчислити об'єм зрізаної піраміди, основами якої є квадрати зі сторонами a та b . $V = h(S_1 + \sqrt{S_1 S_2} + S_2)/3$; S_1, S_2 – площі основ, h – висота піраміди. Значення a, b, h ввести з екрана
25	Обчислити рентабельність роботи підприємства за місяць за формулою $\text{рентабельність} = \text{прибуток} / \text{собівартість} \cdot 100 \%$, якщо собівартість продукції в поточному місяці зменшилась порівняно з минулим на 2 %. Значення прибутку і собівартості за минулий місяць ввести з екрана
26	Обчислити хвильовий опір напівхвильового вібратора $p = 120(\ln(2\lambda/(\pi d)) - 0,577)$, $\lambda = (3 + 0,1 n)$. Значення n та d ввести з екрана
27	Знайти радіуси описаного R і вписаного r кіл для правильного n -кутника з числом сторін n і довжиною сторони a . $R = a/(2\sin(\pi/n))$, $r = a/\text{tg}(\pi/n)$. Значення n і a ввести з екрана
28	Обчислити об'єм зрізаного конуса, основи якого мають радіуси R та r . $V = h(S_1 + \sqrt{S_1 S_2} + S_2)/3$; S_1, S_2 – площі основ, h – висота конуса. Значення R, r, h ввести з екрана
29	Ввести координати точки площини (x, y) . Здійснити перехід до полярних координат (ρ, ϕ) , де $\rho = \sqrt{x^2 + y^2}$, $\text{tg } \phi = y/x$
30	Тіло рухається за законом $S = t^3 - 3t^2 + 2$. Обчислити швидкість тіла в момент часу t . Значення t ввести з екрана. (Функція швидкості є похідною від функції відстані)

Лабораторна робота № 3

Програмування розгалужень.

Умовний оператор **if**

Мета роботи: набути практичних навиків використання умовного оператора *if* при створюванні програмних проектів розгалуженої структури в C++.

Теоретичні відомості

1. Розгалужені алгоритми

Розгалуженням називається вибір програмою тієї чи іншої низки команд залежно від того, чи виконується певна умова. При цьому спрацьовує лише одна з гілок алгоритму.

Для програмної реалізації таких обчислень слід використовувати оператори передавання керування, котрі дозволяють змінювати порядок виконання операторів програми. У мові C++ для цього передбачено інструкції: безумовного переходу – **goto**, умовного переходу – **if** та вибору варіанта – **switch**. Для записування умови переходу слід використовувати логічні (булеві) вирази.

2. Операції відношення та логічні операції

Логічні змінні можуть набувати значень: **true** (істина) або **false** (хибність). У C++ здебільшого ці значення позначають цифрами 1 та 0 (1 – істина, 0 – хибність). Логічний тип ще називають *булевим*, від прізвища англійського математика Джорджа Буля, засновника математичної логіки. При оголошенні змінних булевого типу їх позначають як **bool**, наприклад:

```
bool m;
```

Операції відношень (<, >, <=, >=, == (дорівнює), != (не дорівнює)) порівнюють два вирази і видають значення 1 (**true** – істина – “так”) або 0 (**false** – хибність – “ні”). Типом результату є **int** (або **bool**).

Логічна операція – дія, яка виконується над логічними змінними, її результат є 1 (**true**) або 0 (**false**). Логічні операції обчислюють кожен операнд з огляду на його еквівалентність нулю.

Базові логічні операції:

|| – *логічне додавання* (операція “АБО”, *диз’юнкція*), результатом виконання якого є значення 1 (**true**), якщо хоча б один з операндів має ненульове значення. Якщо перший операнд має ненульове значення, то другий операнд вже не обчислюється;

&& – *логічне множення* (операція “І”, *кон’юнкція*), результатом виконання якого є 1, якщо обидва операнди мають ненульові значення, у решті випадків результат – 0. Якщо значення першого операнда дорівнює 0, то другий операнд не обчислюється;

! – *логічне заперечення* (операція “НЕ”, *інверсія*) виконується над однією логічною змінною, результатом чого є значення **true** (1), якщо початковим значенням було **false** (0), і **false** (0), – якщо початковим значенням було **true** (1).

3. Умовний оператор if

Оператор `if` має дві форми: скорочену та повну.

Скорочена форма має вигляд:

if (<умова>) <оператор>;

Якщо значення логічного виразу в умові є ненульове, тобто умова є істинна, то виконується оператор чи група операторів в операторних дужках {}, інакше відбудеться перехід на наступний оператор.

Повна форма цього оператора:

if (<умова>) <оператор1>;
else <оператор2>;

Якщо значення логічного виразу в умові є ненульове, тобто умова є істинна, то виконуватиметься оператор1 чи група операторів в операторних дужках {}, інакше – виконуватиметься оператор2, після чого відбудеться перехід на наступний оператор. Зауважимо, що обчислюється лише один з операторів, а не обидва.

Наприклад, обчислення значення виразу $y = \begin{cases} 1 + b^x & \text{за } x = b; \\ \frac{x + b}{b - x} & \text{за } x \neq b \end{cases}$ можна реалізувати чи то двома операторами `if` скороченої форми, чи одним оператором `if` повної форми:

- 1) `if(x == b) y = 1 + pow(b, x);`
`if(x != b) y = (x + b)/(b - x);`
- 2) `if(x == 0) y = 1 + pow(b, x); else y = (x + b)/(b - x);`

Приклади програм з розгалуженою структурою в C++

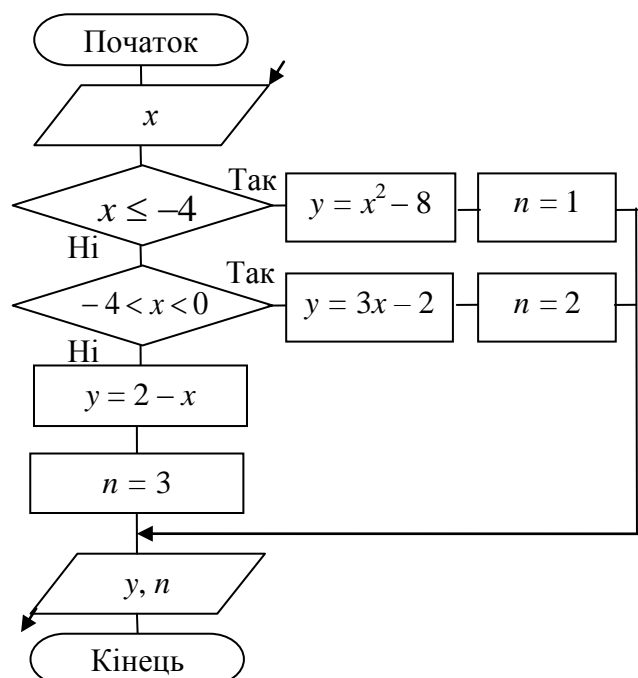
Приклад 1. Ввести x та обчислити

$$y = \begin{cases} x^2 - 8 & \text{за } x \leq -4; \\ 3x - 2 & \text{за } -4 < x < 0; \\ 2 - x & \text{за } x \geq 0. \end{cases}$$

Текст програми та схема алгоритму:

```
#include <iostream>
using namespace std;

int main ()
{
    setlocale(0, ".1251");
    double y, x; int n;
    cout<<"Введіть значення x=";
    cin>>x;
    if(x <= -4) { y=x*x-8; n=1;}
    else
```



```

    if(-4<x && x<0){ y=3*x-2; n=2;}
    else { y=2-x; n=3;}
    cout<<"Результат у="<< y <<endl;
    cout<<"Виконано умову "<<n<<endl;
    system ("pause>>void");
    return 0;
}

```

Результати роботи:

Введіть значення $x=0.4$

Результат $y=1.6$

Виконано умову 3

Приклад 2. Ввести x та обчислити

$$y = \begin{cases} \cos^{3.5}(a+xz) + e^{|bx|} & \text{за } |1-x^2| = a+z; \\ z + \ln|a+bx| & \text{за } |1-x^2| > a+z; \\ \sqrt{ab^4 + \sqrt[5]{zx^2}} & \text{за } |1-x^2| < a+z, \end{cases}$$

де $a = 0.3$; $b = 1.25$; $z = (x+b)^2$.

Текст програми та схема алгоритму:

```

#include <iostream>
using namespace std;
int main ()
{
    setlocale(0, ".1251");
    double a=0.3, b=1.25, x, y, z;
    cout<<"Введіть значення x= ";
    cin>>x;
    z = pow(x + b, 2);
    if (fabs(1-x*x) == a+z) y = pow(cos(a+x*z),3.5) + exp(fabs(b*x));
    else
        if (fabs(1-x*x) > a+z) y = z+log(fabs(a+b*x));
        else y = sqrt(a * pow(b,4) + pow(z*x*x,1./5));
    cout<<"\nРезультат у = "<< y <<endl;
    system ("pause>>void");
    return 0;
}

```

Результати роботи:

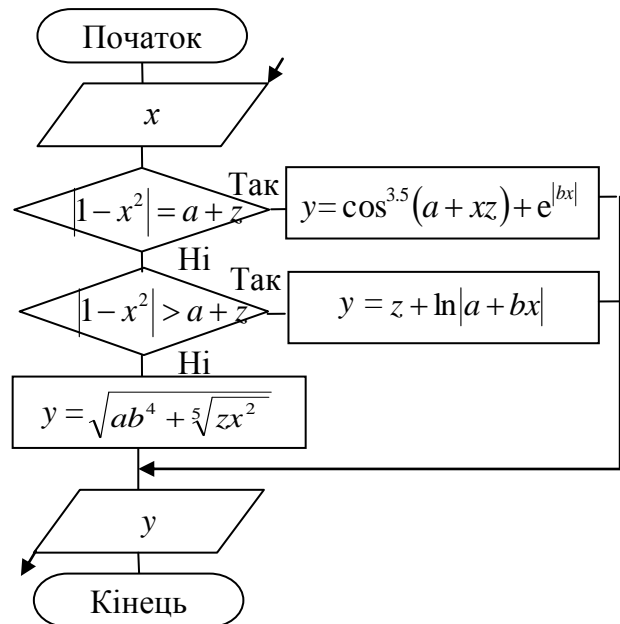
Введіть значення $x = 2.45$

Результат $y = 1.77414$

Приклад 3. Ввести координати точки B (змінні x та y) та визначити, чи ле-

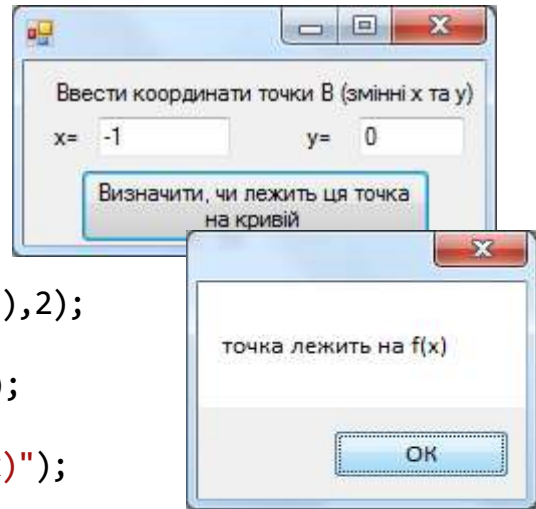
жить ця точка на кривій $f(x) = \begin{cases} -e^x + x^4 & \text{за } |x| > 1; \\ \arccos^2 x^4 & \text{за } |x| \leq 1. \end{cases}$ Похибку задати

$\text{eps} = 10^{-3}$ (тобто $|f(x) - y| < \text{eps}$). Відповідь вивести у вигляді повідомлення.




Текст програми:

```
private: System::Void button1_Click(System::Object^sender, System::EventArgs^e)
{ double x, y, fx;
  const double eps = 1e-3;
  x = Convert::ToDouble(textBox1->Text);
  y = Convert::ToDouble(textBox2->Text);
  if (Math::Abs(x) > 1)
    fx = -Math::Exp(x) + Math::Pow(x, 4);
  else
    fx = Math::Pow(Math::Acos(Math::Pow(x,4)),2);
  if (Math::Abs(fx - y) < eps)
    MessageBox::Show("точка лежить на f(x)");
  else
    MessageBox::Show("точка не лежить на f(x)");
}
```



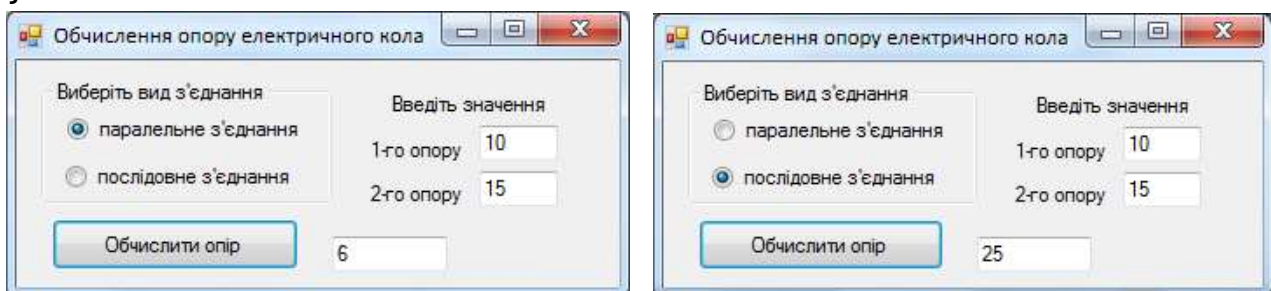
Приклад 4. Створити програму для обчислення опору електричного кола, який складається з двох з'єднаних опорів, залежно від виду їхнього з'єднання: паралельне або послідовне (формули див. у прикладі 3 в лабораторній роботі № 2).

Тобто вдосконалимо розв'язок прикладу 3 в лабораторній роботі № 2 за рахунок можливості вибору виду з'єднаних опорів. Для цього використаємо елемент `groupBox`, на якому розмістимо два елементи `radioButton`. Для `radioButton1` слід встановити значення `Checked` в `True`. Перемикачі `radioButton` дозволяють "увімкнути" лише одну з усіх радіокнопок , при цьому решта кнопок автоматично вимикається.

Програмне опрацювання вибору кнопки можна реалізувати оператором `if-else`.

Текст програми:

```
private: System::Void button1_Click(System::Object^sender, System::EventArgs^e)
{ double r1, r2, R;
  r1 = Convert::ToDouble(textBox1->Text); // значення 1-го опору r1
  r2 = Convert::ToDouble(textBox2->Text); // значення 2-го опору r2
  if(radioButton1->Checked) R = r1*r2/(r1+r2);
  else R = r1+r2;
  textBox3->Text = Convert::ToString(R);
}
```



Питання для самоконтролю

- 1) Який процес називають розгалуженням?
- 2) Які оператори в C++ використовуються для організації розгалужень?
- 3) Перелічіть базові логічні операції.
- 4) Обчисліть логічний вираз $(-3 \geq 5) \parallel (7 < 9) \&\& (0 < 3)$.
- 5) Вкажіть значення w після виконання оператора `bool w=2*5<=17%3;`.
- 6) Запишіть умовні оператори `if` скороченої і повної форми для обчис-

$$\text{лення } y = \begin{cases} \sin x^2 & \text{за } x > 0.5; \\ \cos^2 x & \text{за } x \leq 0.5. \end{cases}$$

- 7) Назвіть оператори без помилок:
 - а) `if(x<=6)y=2*x; else y=cos(x);` в) `if(a<>0) if(b<>0) y=2*x;`
 - б) `if y<=x then y:=exp(x*y);` г) `if(x>0)y=ln(x) else y=x;`
- 8) Вкажіть значення x після виконання фрагментів програми:
 - а) `float x=1.5;` б) `float x=1.5;`
 - `if(x<=0.5) x=7.7;` `if(x<=0.5) x=7.7; else x=3;`
- 9) Наведіть значення змінної z після виконання операторів:


```
float z, x=2.5;
if(x>=0.5) z=7.7; else z=5.5;
```

Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
- 2) У протоколі лабораторної роботи скласти схеми алгоритмів і написати програми мовою C++ із застосуванням умовного оператора `if` для розв'язання завдань, поданих в табл. 3.1 ... 3.3 відповідно до індивідуального варіанта.
- 3) Створити на комп'ютері програмні проекти в середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Таблиця 3.1

Індивідуальні завдання базового рівня складності

№ вар.	Функція	№ вар.	Функція
1	$Y = \begin{cases} x^2 + 1 & \text{за } x < 0; \\ x^2 - 1 & \text{за } 0 \leq x < 2; \\ x & \text{за } x \geq 2 \end{cases}$	2	$Y = \begin{cases} 2x + 2 & \text{за } x < -3; \\ 2x - 2 & \text{за } -3 \leq x \leq 0; \\ x^2 & \text{за } x > 0 \end{cases}$
3	$Y = \begin{cases} 6x + 8 & \text{за } x \leq -5; \\ x - 2 & \text{за } -5 < x \leq 3; \\ 2x^2 & \text{за } x > 3 \end{cases}$	4	$Y = \begin{cases} 2x - 1 & \text{за } x \leq -4; \\ x^2 + 2 & \text{за } -4 < x \leq 5; \\ x + 3 & \text{за } x > 5 \end{cases}$
5	$Y = \begin{cases} 6x^3 - 8 & \text{за } x \leq -8; \\ x^3 - 8 & \text{за } -8 < x < 0; \\ 2x^2 & \text{за } x \geq 0 \end{cases}$	6	$Y = \begin{cases} 2x^3 + 3x & \text{за } x \leq -1; \\ x^2 - 4 & \text{за } -1 < x < 0; \\ x^3 & \text{за } x \geq 0 \end{cases}$

Продовження табл. 3.1

№ вар.	Функція	№ вар.	Функція
7	$Y = \begin{cases} 4x^2 + 2x & \text{за } x \leq -12; \\ 2x^2 + 2x & \text{за } -12 < x < 3; \\ x + 1 & \text{за } x \geq 3 \end{cases}$	8	$Y = \begin{cases} x^3 - 1 & \text{за } x \leq -4; \\ 2x - 1 & \text{за } -4 < x \leq 3; \\ 3x^3 & \text{за } x > 3 \end{cases}$
9	$Y = \begin{cases} 4x + 3 & \text{за } x \leq -2; \\ 2x^2 - 4 & \text{за } -2 < x < 4; \\ x^2 - 2 & \text{за } x \geq 4 \end{cases}$	10	$Y = \begin{cases} 2x + 4 & \text{за } x \leq -1; \\ x - 4 & \text{за } -1 < x < 0; \\ x^3 + 4 & \text{за } x \geq 0 \end{cases}$
11	$Y = \begin{cases} 4x^2 + 2x & \text{за } x < -2; \\ 2x - 1 & \text{за } -2 \leq x < 3; \\ x^3 + 3 & \text{за } x \geq 3 \end{cases}$	12	$Y = \begin{cases} 3x^2 + 2x & \text{за } x < -3; \\ 2x + 1 & \text{за } -3 \leq x < 8; \\ 3x & \text{за } x \geq 8 \end{cases}$
13	$Y = \begin{cases} 4x + 2x & \text{за } x \leq -4; \\ x - 2x & \text{за } -4 < x < 2; \\ x + 2 & \text{за } x \geq 2 \end{cases}$	14	$Y = \begin{cases} 27x + 3 & \text{за } x \leq -6 \\ x^3 - 1 & \text{за } -6 < x < 3 \\ x^2 + 1 & \text{за } x \geq 3 \end{cases}$
15	$Y = \begin{cases} x^3 + 2x^2 & \text{за } x \leq -2; \\ x^2 - 1 & \text{за } -2 < x < 3; \\ 2x + 2 & \text{за } x \geq 3 \end{cases}$	16	$Y = \begin{cases} 4x^3 + 2x & \text{за } x < -4; \\ 2x - 5 & \text{за } -4 \leq x < 4; \\ x - 3 & \text{за } x \geq 4 \end{cases}$
17	$Y = \begin{cases} 6x^2 + 2x & \text{за } x \leq -6; \\ 2x - 6 & \text{за } -6 < x < 4; \\ 6x + 1 & \text{за } x \geq 4 \end{cases}$	18	$Y = \begin{cases} 27x^2 + 1 & \text{за } x \leq -3; \\ x - 2 & \text{за } -3 < x < 5; \\ 3x + 1 & \text{за } x \geq 5 \end{cases}$
19	$Y = \begin{cases} 8x^3 + 2 & \text{за } x \leq -1; \\ x^2 - 1 & \text{за } -1 < x < 1; \\ x + 1 & \text{за } x \geq 1 \end{cases}$	20	$Y = \begin{cases} 21 - x & \text{за } x \leq -7; \\ x^2 + 3 & \text{за } -7 < x < 4; \\ x^2 - 3 & \text{за } x \geq 4 \end{cases}$
21	$Y = \begin{cases} 2x^2 + 3 & \text{за } x < -2; \\ x^3 - 6 & \text{за } -2 \leq x < 0; \\ 2(x + 1) & \text{за } x \geq 0 \end{cases}$	22	$Y = \begin{cases} 4x^3 + 4 & \text{за } x \leq -2; \\ 3x - 3 & \text{за } -2 < x \leq 3; \\ 2x_2 + 2 & \text{за } x > 3 \end{cases}$
23	$Y = \begin{cases} x^3 + 2x & \text{за } x \leq -3; \\ 2x - 1 & \text{за } -3 < x \leq 8; \\ x^2 + 1 & \text{за } x > 8 \end{cases}$	24	$Y = \begin{cases} 25x + 1 & \text{за } x \leq -2; \\ x^3 - 25 & \text{за } -2 < x < 4; \\ 24x + x^2 & \text{за } x \geq 4 \end{cases}$
25	$Y = \begin{cases} 26x + 4 & \text{за } x \leq -6; \\ 4x^2 + 2 & \text{за } -6 < x < 6; \\ 2x - 3 & \text{за } x \geq 6 \end{cases}$	26	$Y = \begin{cases} 9x^3 + 1 & \text{за } x \leq -9; \\ x^2 - 1 & \text{за } -9 < x \leq 1; \\ x + 2 & \text{за } x > 1 \end{cases}$

Закінчення табл. 3.1

№ вар.	Функція	№ вар.	Функція
27	$Y = \begin{cases} 4x^2 + 4 & \text{за } x \leq -4; \\ x^3 - 1 & \text{за } -4 < x < 4; \\ x^2 + 1 & \text{за } x \geq 4 \end{cases}$	28	$Y = \begin{cases} x^3 - 29 & \text{за } x \leq -3; \\ 2x + 3 & \text{за } -3 < x \leq 6; \\ x^2 + 1 & \text{за } x > 6 \end{cases}$
29	$Y = \begin{cases} 3x + 1 & \text{за } x \leq -3; \\ x^2 - 1 & \text{за } -3 < x < 4; \\ x^3 + 1 & \text{за } x \geq 4 \end{cases}$	30	$Y = \begin{cases} 2x^3 + 4x & \text{за } x \leq -1; \\ x + 4 & \text{за } -1 < x < 3; \\ 2x + 2 & \text{за } x \geq 3 \end{cases}$

Таблиця 3.2

Індивідуальні завдання середнього рівня складності

№ вар.	Функція	№ вар.	Функція
1	$y = \begin{cases} a^3 + \arcsin(\cos^3 bx) & \text{за } x \leq a; \\ \sqrt{(a+bx)-2} + \sin x & \text{за } a < x < b; \\ \operatorname{tg}^2(a+bx+z) & \text{за } x \geq b, \end{cases}$ де $a = 2.5; b = 3.5; z = \sin(bx)$	2	$y = \begin{cases} a^{2b}x^2 + \sqrt{b^4 + 2.7} & \text{за } x < 0.7; \\ \operatorname{arctg}(3^x - px) & \text{за } x = 0.7; \\ \sqrt[3]{\ln a-px + 4.3} & \text{за } x > 0.7, \end{cases}$ де $a = 0.54; b = 0.34; p = ax + b$
3	$y = \begin{cases} (a+z)\cos^2(bx+x^3) & \text{за } x < a; \\ a \ln(zx) + \sin^2(b^2) & \text{за } a \leq x \leq b; \\ \sqrt[3]{0.3b + \sqrt{(a-z^2)}} & \text{за } x > b, \end{cases}$ де $a = 0.1; b = 3.25; z = \cos^2(x)$	4	$y = \begin{cases} \operatorname{ctg}(x^2e^{3k}) + \ln rx & \text{за } x = rs; \\ \sqrt[5]{x^2} + \sqrt{ \sin k } & \text{за } x > rs; \\ \operatorname{tg}(kx + \operatorname{tg}(rs)) & \text{за } x < rs, \end{cases}$ де $r = 2.4; s = 5; k = 0.5$
5	$y = \begin{cases} \frac{(2a+1)^2}{3.71-x^2} & \text{за } z > -0.5; \\ \sin^3 \sqrt{bz} - ax & \text{за } -0.5 \leq z \leq 10^{-3}; \\ \frac{\operatorname{tg}(z+x) - e^x}{3.5abx} & \text{за } z > 10^{-3}, \end{cases}$ де $a = 0.3; b = 0.7; z = \cos(x+2)$	6	$y = \begin{cases} e^{ax} + f \cos^5 bx & \text{за } x \leq a; \\ \cos^2 \sqrt{bx} - \ln(fx) & \text{за } a < x \leq b^2; \\ \cos^2(a+bfx) & \text{за } x > b^2, \end{cases}$ де $a = 1.5; b = 1.44; f = \sqrt{bx}$
7	$y = \begin{cases} a \cos^2 x + b \sin^2 zx & \text{за } x \leq a; \\ a \cdot \operatorname{tg}(\sin^2 bx + z) & \text{за } a < x \leq 4.5b; \\ \ln(ax-b) + z^2 & \text{за } x > 4.5b, \end{cases}$ де $a = 1.5; b = 0.7; z = \operatorname{tg} \operatorname{tg}(bx) $	8	$y = \begin{cases} \ln bzx + za^{2.5} & \text{за } a^3 < x \leq b; \\ ax^2 + bz^a + \sin^2 zx & \text{за } x \leq a^3; \\ \cos(ax+b) + \ln zx & \text{за } x > b, \end{cases}$ де $a = 0.5; b = 0.7; z = 0.2$

Продовження табл. 3.2

№ вар.	Функція	№ вар.	Функція
9	$y = \begin{cases} \sin(e^{a+b}) + x^2 & \text{за } a + b > x; \\ \operatorname{arctg}(abc) + \sqrt[3]{x} & \text{за } a + b = x; \\ \operatorname{arcsin}(\cos^2(\sqrt{ x })) & \text{за } a + b < x, \end{cases}$ де $a = 0.5; b = 1.5; c = 3.2$	10	$y = \begin{cases} \ln(\lg kx + mn) & \text{за } x > m + n ; \\ \sin(kmx) + \sqrt{ nx } & \text{за } x = m + n ; \\ e^{\cos x} + e^{m+n} & \text{за } x < m + n , \end{cases}$ де $m = 2.1; n = 1.9; k = 8.5$
11	$y = \begin{cases} a \sin^2 x + \cos(zx) & \text{за } x < \ln(a); \\ \cos^3(a + zx) & \text{за } \ln(a) \leq x \leq b; \\ \sqrt{2.5a^3 + (b - zx^2)^6} & \text{за } x > b, \end{cases}$ де $a = 0.1; b = 3.25; z = \cos^2(x)$	12	$y = \begin{cases} \sin(bm + \cos(nx)) & \text{за } bm > x^2; \\ \cos(bm - \sin x) & \text{за } bm < x^2; \\ \sqrt{e^{ \cos x } + \sqrt{ bmx }} & \text{за } bm = x^2, \end{cases}$ де $m = 0.5; b = -2; n = 0.2$
13	$y = \begin{cases} xe^x + (z + 7.7abx) & \text{за } x < a; \\ \operatorname{tg}(ax + z) + \cos^2 bx & \text{за } a \leq x \leq b^2; \\ \ln(\sin^2(a + bx + zx^2)) & \text{за } x > b^2. \end{cases}$ де $a = 1.5; b = -1.7; z = 1.2$	14	$y = \begin{cases} c \sin^2(b^2 x) + \ln(cx + a) & \text{за } x < a; \\ a + \ln(bx) - \sin(cx) & \text{за } a \leq x < b; \\ \sqrt{ \cos(a + bx) + cx^2 } & \text{за } x \geq b, \end{cases}$ де $a = 0.5; b = 0.7; c = 3.4$
15	$y = \begin{cases} a \sin^2 x + b \cos(zx) & \text{за } x < a^3; \\ (a + bx)^2 - \sin(zx) & \text{за } a^3 \leq x \leq b; \\ \sqrt{ x - \sin(bx + z) } & \text{за } x > b, \end{cases}$ де $a = 0.9; b = 1.25; z = x^2$	16	$y = \begin{cases} 2.8 \sin^2 ax - bx^3 z & \text{за } x < a; \\ z \cos(ax + b)^2 & \text{за } a \leq x \leq b^2; \\ e^{ 2.5ax } + zabx & \text{за } x > b^2, \end{cases}$ де $a = 0.7; b = -1.25; z = 3.5$
17	$y = \begin{cases} a + \sin bx + \cos x^2 & \text{за } x \leq a; \\ \sqrt{a + bx} + \sin zx & \text{за } a < x < \ln b; \\ \ln(a + bx + z) & \text{за } x \geq \ln b, \end{cases}$ де $a = 0.2; b = 12.5; z = (2.5 + b)^2$	18	$y = \begin{cases} \sin(e^{a+b}) + x^2 & \text{за } e^{a+b} > e^x; \\ \operatorname{arctg}(abz) + \sqrt[3]{x} & \text{за } e^{a+b} = e^x; \\ \cos(\sqrt{ x + abz }) & \text{за } e^{a+b} < e^x, \end{cases}$ де $a = 0.7; b = 2.3; z = (x + b)^2$
19	$y = \begin{cases} \ln^2(0.5a - u) & \text{за } u < -0.5; \\ \frac{2x - a}{7\pi + x + u} & \text{за } -0.5 \leq u \leq 10^{-3}; \\ \lg(u + x) - e^x & \text{за } u > 10^{-3}, \end{cases}$ де $a = 2.3; u = \sin(x + a)$	20	$y = \begin{cases} xe^a + e^{ bc } & \text{за } 1 - x^2 = a + c; \\ \sin^2 ax + \cos bc & \text{за } 1 - x^2 > a + c; \\ \sqrt{ab^4 + \sqrt[5]{cx^2}} & \text{за } 1 - x^2 < a + c, \end{cases}$ де $a = 0.3; b = 0.7; c = 2.7 - x$
21	$y = \begin{cases} x^2 e^{2k} + \ln rx & \text{за } \cos x = \cos(rs); \\ \sqrt[3]{x^2} + \sqrt{ k + rsx } & \text{за } \cos x > \cos(rs); \\ \operatorname{arctg}(kx + rs) & \text{за } \cos x < \cos(rs), \end{cases}$ де $s = 0.15; r = 10; k = 0.7$	22	$y = \begin{cases} \sqrt[3]{b + \sqrt{ x + c }} & \text{за } \lg a < x; \\ \cos(x - b - c) & \text{за } \lg a = x; \\ \sin(x + a - b) & \text{за } \lg a > x, \end{cases}$ де $a = 10; b = 0.7; c = 3.7$
23	$y = \begin{cases} a + bx + \sin^2 zx^{3.5} & \text{за } x < a; \\ a + \ln ab - zx & \text{за } a \leq x \leq b; \\ \sqrt{ a + \operatorname{ctg}(zx) } + bx & \text{за } x > b, \end{cases}$ де $a = 0.2; b = 0.9; z = 1.7$	24	$y = \begin{cases} z^2 - \cos^2(x) & \text{за } x < 3.5a; \\ (a - x)^2 - bx & \text{за } 3.5a \leq x \leq b; \\ \sqrt{\sqrt{bx} + zx^2} & \text{за } x > b, \end{cases}$ де $a = 0.1; b = 2.1; z = abx$

Закінчення табл. 3.2

№ вар.	Функція	№ вар.	Функція
25	$y = \begin{cases} 3.5 \sin^2(bx + z)^3 & \text{за } x \leq a; \\ \ln(a + b^3 x) + a & \text{за } a < x \leq b^{2.5}; \\ \cos^2(a^b + xz) & \text{за } x > b^{2.5}, \end{cases}$ де $a = 0.3; b = 1.7; z = 2$	26	$y = \begin{cases} (3.5a - 7.3bx)^3 & \text{за } x < -\ln a ; \\ a^b - \cos^3(zx) & \text{за } -\ln a \leq x < b; \\ \sqrt{ \operatorname{tg} a - x } - x & \text{за } x \geq b, \end{cases}$ де $a = 0.3; b = 3.7; z = 12.7/x$
27	$y = \begin{cases} \ln mx + n & \text{за } x^2 > m + n; \\ e^{\cos mx - n } & \text{за } x^2 = m + n; \\ \sqrt[3]{k^2 + \cos^2 x} & \text{за } x^2 < m + n, \end{cases}$ де $m = 2.1; n = 1.9; k = 3.5$	28	$y = \begin{cases} 2.5b^2 + ax - \cos xz & \text{за } x \leq 5a; \\ (a^2 - x)^3 + \ln(xz) & \text{за } x > b; \\ \sqrt{b^2 + (a - x^3 z)^2} & \text{за } 5a < x \leq b, \end{cases}$ де $a = 0.3; b = 1.7; z = 0.7$
29	$y = \begin{cases} \sqrt{ a - \cos^2 b^3 x + c^2 } & \text{за } 1 - x^2 = a + c; \\ e^{0.04x} + \ln b^5 \cos x & \text{за } 1 - x^2 > a + c; \\ \cos^2(b^3 x) + \ln bx - a & \text{за } 1 - x^2 < a + c, \end{cases}$ де $a = 0.7; b = 1.25; c = 1.3$	30	$y = \begin{cases} e^{ax} - 3.5 \cos^2(z + bx) & \text{за } x \leq a; \\ a + \ln a + bx - 2x & \text{за } a < x \leq b; \\ a + \cos^{3.5}(a + bxz) & \text{за } x > b, \end{cases}$ де $a = 0.7; b = 1.25; z = (x + b)^2$

Таблиця 3.3

Індивідуальні завдання високого рівня складності

№ вар.	Завдання
1	Ввести два числа і визначити, що більше: сума квадратів чи квадрат суми цих чисел. Відповідь вивести у вигляді повідомлення
2	Ввести значення кута в радіанах і визначити, що більше: значення синуса чи косинуса цього кута. Відповідь вивести у вигляді повідомлення
3	Ввести три числа і визначити серед них середнє за значенням
4	Ввести три числа і визначити серед них найменше
5	Ввести координати точки $B(x \text{ та } y)$ і визначити: чи належить ця точка кривій $f(x) = 6x^7 - 4.5x^5 + 4x^2$ з допустимою похибкою $\text{eps} = 10^{-3}$ (тобто $ f(x) - y < \text{eps}$)
6	Ввести координати точок $A(x_0, y_0)$ і $B(x_1, y_1)$ і визначити яка з цих точок – А чи В – є найбільш віддалена від початку координат $O(0,0)$. Відповідь вивести у вигляді повідомлення
7	Ввести значення трьох сторін трикутника a, b та c і визначити, чи є цей трикутник прямокутним. Відповідь вивести у вигляді повідомлення
8	Ввести три числа і додатні з них піднести до квадрата, а від'ємні залишити без змін
9	Ввести координати точки $A(x \text{ та } y)$ і визначити: в якій чверті лежить ця точка. Відповідь вивести у вигляді повідомлення
10	Ввести координати точки x і y та визначити, чи лежить ця точка всередині кола з радіусом R . Центром кола є початок координат. Відповідь вивести у вигляді повідомлення

Закінчення табл. 3.3

№№	Завдання
11	Ввести три цілих числа (довжини сторін трикутника) і визначити, чи можна побудувати за цими числами трикутник
12	Ввести значення сторони квадрата A та радіус кола R і визначити, площа якої з цих фігур є більше. Відповідь вивести у вигляді повідомлення
13	Ввести значення трьох сторін двох трикутників – a_1, b_1, c_1 і a_2, b_2, c_2 . Визначити, площа якого з них є більше. Відповідь вивести у вигляді повідомлення
14	Ввести координати точки B (x та y) і визначити: чи належить ця точка кривій $f(x) = 6\cos^2 x - 0.25x^5 + 3.2x^2 - 2.7$ з припустимою похибкою $\text{eps} = 10^{-3}$ (тобто $ f(x) - y < \text{eps}$). Відповідь вивести у вигляді повідомлення
15	Ввести три числа і додатні з них піднести до куба, а від'ємні – замінити на 0
16	Ввести значення трьох сторін трикутника a, b , і c . Найменша зі сторін трикутника є стороною квадрата. Визначити, площа якої з цих фігур є більша
17	Ввести координати точки $A(x$ та $y)$ і визначити, чи належить ця точка до першої чверті. Відповідь вивести у вигляді повідомлення
18	Ввести три числа і вивести числа за модулем, більші за середнє арифметичне
19	Ввести радіанну міру кута і визначити більше зі значень тангенса або котангенса цього кута. Відповідь вивести у вигляді повідомлення
20	Ввести координати точки $Q(x$ та $y)$ і визначити, чи лежить ця точка на кривій $f(x) = 7\text{tg}^2 x - 0.31x^3 + 3.2x^2 - e^x$ з припустимою похибкою $\text{eps} = 10^{-3}$ (тобто $ f(x) - y < \text{eps}$). Відповідь вивести у вигляді повідомлення
21	Ввести три числа і визначити найбільше з них
22	Ввести два числа і визначити, що є більше: різниця квадратів чи модуль квадрата різниці цих чисел. Відповідь вивести у вигляді повідомлення
23	Ввести координати точок $A(x_0, y_0)$ та $B(x_1, y_1)$ і визначити, яка з точок – A чи B – найменш віддалена від початку координат $O(0,0)$. Відповідь вивести у вигляді повідомлення
24	Ввести координати точки $A(x$ та $y)$ і визначити, чи лежить ця точка всередині тора, утвореного колами із радіусами r і R із центром в точці $O(0,0)$
25	Ввести координати точки $B(x$ та $y)$ і визначити, чи лежить ця точка на кривій $f(x) = \begin{cases} \sin^2 x^3 & \text{за } x > 1; \\ \sqrt{6\arcsin x^7 + 4.5x^6 + 4x^2 + 2} & \text{за } x \leq 1 \end{cases}$ із припустимою похибкою $\text{eps} = 10^{-3}$ (тобто $ f(x) - y < \text{eps}$). Відповідь вивести у вигляді повідомлення
26	Ввести координати точки $A(x$ та $y)$ і визначити, чи лежить ця точка в четвертій чверті. Відповідь вивести у вигляді повідомлення
27	Ввести значення трьох сторін трикутника a, b та c і визначити, чи є цей трикутник рівнобедреним. Відповідь вивести у вигляді повідомлення
28	Ввести три цілих числа a, b, c і визначити, чи є вони трійкою Піфагора ($c^2 = a^2 + b^2$). Відповідь вивести у вигляді повідомлення.
29	Ввести координати точки $A(x$ та $y)$ і визначити, чи лежить ця точка в області, обмеженій параболою $y = 2 - x^2$ та віссю абсцис
30	Ввести координати точок $A_1(x_1, y_1)$, $A_2(x_2, y_2)$, $A_3(x_3, y_3)$ і визначити, чи лежать ці точки на одній прямій. Відповідь вивести у вигляді повідомлення

Лабораторна робота № 4

Програмування розгалужень.

Оператор вибору варіантів **switch**

Мета роботи: набути практичних навиків використання операторів вибору варіантів **switch** та безумовного переходу **goto**, а також умовної операції **?:** при створенні програмних проектів розгалуженої структури в C++.

Теоретичні відомості

1. Оператор безумовного переходу **goto**

Оператор **goto** (перейти до) дозволяє передавати керування у будь-яку точку коду (програми), котру позначено спеціальною міткою.

Синтаксис оператора **goto**:

goto <мітка>;

Мітку записують перед оператором, на який слід передати керування, і відокремлюють від нього символом двокрапки (:). Мітки в C++ не оголошують. В якості мітки може застосовуватись поєднання будь-яких літер латиниці і цифр, але розпочинатися мітка повинна з літери, наприклад: **start**, **M1**, **second**.

Застосовувати цей оператор слід обережно і помірковано, особливо при переході всередину блоку чи циклу, оскільки це може призвести до непередбачуваних помилок. Тому в C++ оператор **goto** застосовується рідко і вважається застарілим. Однак, на практиці часто трапляються випадки, коли цей оператор значно спрощує код програми.

2. Умовна операція **?:**

Формат:

<умова> ? <операнд1> : <операнд2>;

Якщо *умова* має ненульове значення, результатом буде значення *операнда1*, інакше – значення *операнда2*. Зауважимо, що обчислюється лише один з операндів, а не обидва.

Наприклад:

j = (i < 0) ? (-i) : (i);

У результаті *j* набуде абсолютного значення *i*, оскільки, якщо *i* є менше нуля, то *j* присвоїться *-i*, а, якщо *i* більше або дорівнює нулю, то *j* присвоїться *i*.

Для визначення більшого з двох чисел *x* та *y* слід записати оператор:

max = (x > y) ? x : y;

Приміром, організувати почергове змінювання тексту на кнопці при клацанні по ній можна лише однією командою:

button1->Text = (button1->Text == "on") ? "off" : "on";

Умовні операції можна вкладати одна в одну. Наприклад, залежно від значення числа можна формувати коректну кінцівку рядка:

```
int k;
cout<<"Введіть значення кількості комп'ютерів - ";
cin>> k;
int n=(k>20)? k%10 : k;
cout<<"В класі є " << k << " комп'ютер"
    << ((!n || n>4) ? "ів." : (n>1) ? "и." : ".")<<endl;
```

```
Введіть значення кількості комп'ютерів - 3
В класі є 3 комп'ютери.
```

```
Введіть значення кількості комп'ютерів - 15
В класі є 15 комп'ютерів.
```

```
Введіть значення кількості комп'ютерів - 21
В класі є 21 комп'ютер.
```

3. Оператор вибору варіантів switch

Формат оператора вибору варіантів switch:

```
switch (<вираз>)
{ case <значення-мітка_1>: {<послідовність операторів>; break;}
  .....
  case <значення-мітка_n>: {<послідовність операторів>; break;}
  [ default: <послідовність операторів>; ]
}
```

Спочатку обчислюється *вираз* у дужках. *Вираз* повинен мати цілий або символьний тип. Значення *виразу* порівнюється зі значеннями *міток* після ключових слів *case*. Якщо значення *виразу* збіглося зі значенням якої-небудь *мітки*, то виконується відповідна *послідовність операторів*, позначена цією *міткою* і записана після двокрапки, доки не зустрінеться оператор *break*. Якщо значення *виразу* не збіглося з жодною *міткою*, то виконуються оператори, які слідують за ключовим словом *default*. Мітка *default* є необов'язковою конструкцією оператора *switch*, на що вказують квадратні дужки *[]* у форматі. Оператор *break* здійснює вихід із *switch*. Якщо оператор *break* є відсутній наприкінці операторів відповідного *case*, то буде по чергову виконано всі оператори до наступного *break* чи то до кінця *switch* для всіх гілок *case* незалежно від значення їхніх міток.

Розглянемо роботу оператора *switch* на прикладі функції, яка по значенню введеного цілого числа виводить назву дня тижня.

```
#include <iostream>
#include <locale.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    int n; char *s="";
    cout << "Введіть ціле число від 1 до 7: "; cin >> n;
```

```
Введіть ціле число від 1 до 7: 5
5-й день тижня - п'ятниця
```

```
Введіть ціле число від 1 до 7: 22
Помилка! Число не є значенням від 1 до 7
```

```

switch (n)
{
    case 1: s="понеділок"; break;
    case 2: s="вівторок"; break;
    case 3: s="середа"; break;
    case 4: s="четвер"; break;
    case 5: s="п'ятниця"; break;
    case 6: s="субота"; break;
    case 7: s="неділя"; break;
    default: { cout<<"\nПомилка! Число не є значенням від 1 до 7\n";
               system("pause>>void");return 0;
            }
}
cout << endl << n << "-й день тижня - " << s << endl;
system("pause>>void");
return 0;
}

```

Наведемо ще один приклад програми з використанням оператора switch для реалізації простого калькулятора в консольному режимі.

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double a,b,res;
    char op;
    cout << "\n Введіть перший операнд "; cin >> a;
    cout << "\n Введіть знак операції "; cin >> op;
    cout << "\n Введіть другий операнд "; cin >> b;
    bool f=true;
    switch (op)
    {
        case '+': res = a+b; break;
        case '-': res = a-b; break;
        case '*': res = a*b; break;
        case '/': res = a/b; break;
        default: cout << "\n Невідома операція!\n "; f=false;
    }
    if (f) cout << "\n Результат : " << res<<endl;
    system("pause>>void");
    return 0;
}

```

```

Введіть перший операнд 0.25
Введіть знак операції *
Введіть другий операнд 50
Результат : 12.5

```

```

Введіть перший операнд 89
Введіть знак операції ^
Введіть другий операнд 10
Невідома операція!

```

Приклади програм з розгалуженою структурою, зорганізованих за допомогою оператора switch

Приклад 1. Обчислити значення функції $L = \begin{cases} x^{2+a} + 1 & \text{за } k = 1; \\ (x+1)/a & \text{за } k = 2; \\ e^{x-a} - a^x & \text{за } k = 3; \\ \lg|x-a| & \text{за } k = 4 \end{cases}$ для всіх

значень параметра k .

Текст програми
та схема алгоритму:

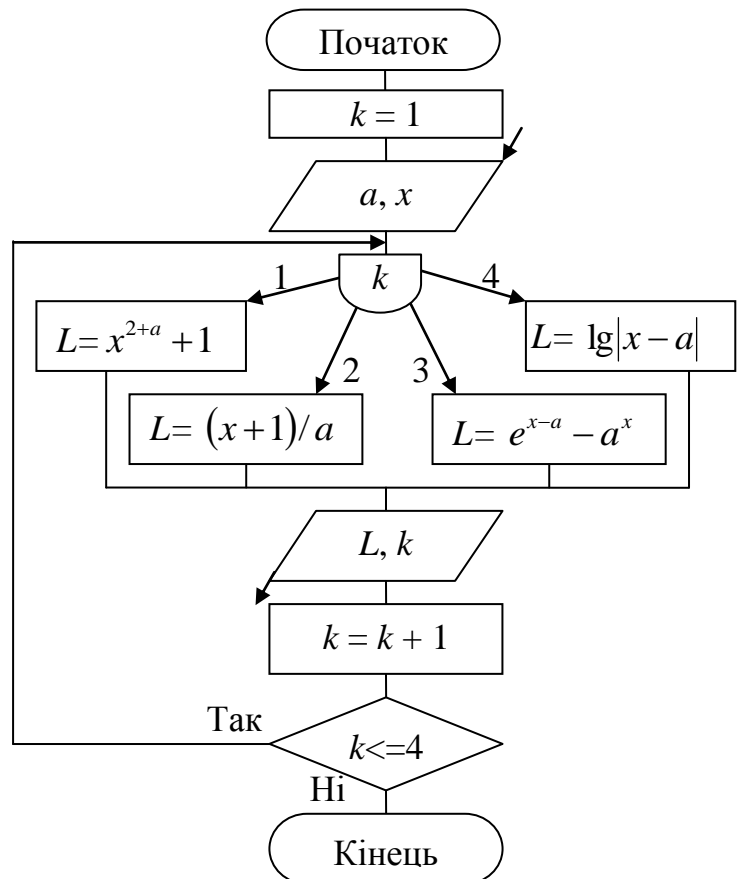
```
#include "stdafx.h"
#include <iostream>
#include <math.h>
using namespace std;
```

```
int main()
{
    setlocale(0, ".1251");
    double x, a, L;
    int k = 1;
    cout<<"Введіть значення a= ";
    cin>> a;
    cout<<"Введіть значення x= ";
    cin>> x;
    cout<< "\nРезультати:\n";
```

M1:

```
    switch (k)
    { case 1: { L = pow(x,2+a) + 1; break; }
      case 2: { L = (x+1)/a; break; }
      case 3: { L = exp(x-a) - pow(a,x); break; }
      case 4: { L = log10(fabs(x-a)); break; }
    }
    cout<< "L" << k << " = " << L << endl;
    k++;
    if(k <= 4) goto M1;
    system("pause>>void");
    return 0;
}
```

Результати роботи:



```
Введіть значення a= 0.8
Введіть значення x= 10.5

Результати:
L1 = 724.319
L2 = 14.375
L3 = 16317.5
L4 = 0.986772
```

Приклад 2. Ввести значення x та обчислити значення

$$y = \begin{cases} \sin(e^{|a+c|}) + x^{2+b} & \text{за } x < a; \\ \sqrt[3]{a + |5.3b|} + c & \text{за } a \leq x < b; \\ \cos^2(a+b) + \sin x^2 & \text{за } x > b \end{cases}$$

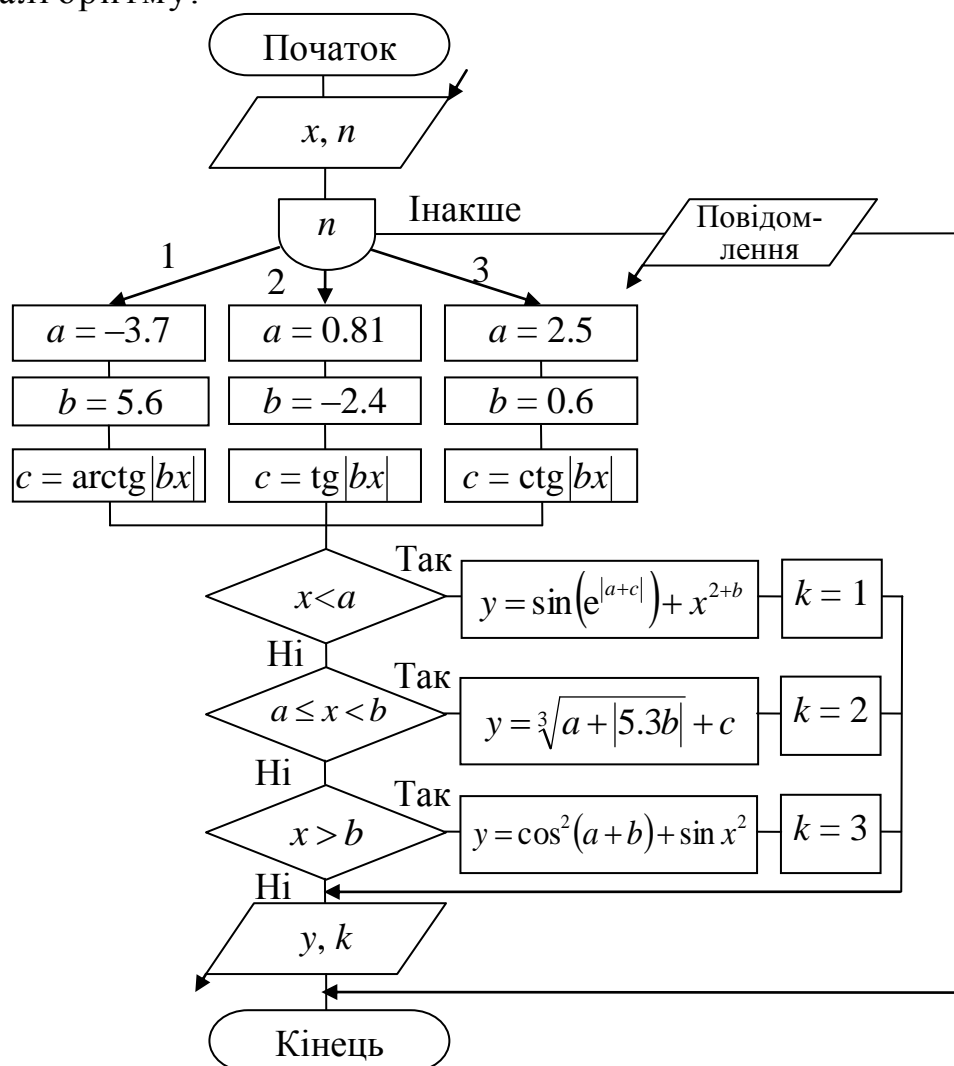
для одного з трьох варіантів параметрів, номер якого слід ввести:

1) $a = -3,7$; $b = 5,6$; $c = \operatorname{arctg}|bx|$;

2) $a = 0,81$; $b = -2,4$; $c = \operatorname{tg}|bx|$;

3) $a = 2,5$; $b = 0,6$; $c = \operatorname{ctg}|bx|$.

Схема алгоритму:



Текст програми:

```

#include "stdafx.h"
#include <iostream>
#include <math.h>
using namespace std;
int main()
{

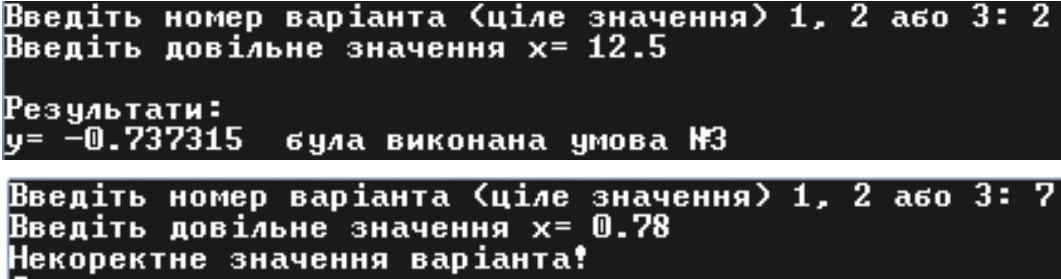
```

```

setlocale(0, ".1251");
double x, y, a, b, c;
int k, n;
cout<<"Введіть номер варіанта (ціле значення) 1, 2 або 3: ";
cin>>n;
cout<<"Введіть довільне значення x= ";
cin>>x;
switch (n)
{ case 1: a=-3.7; b=5.6; c=atan(fabs(b*x)); break;
  case 2: a=0.81; b=-2.4; c=tan(fabs(b*x)); break;
  case 3: a=2.5; b=0.6; c=1/tan(fabs(b*x)); break;
  default: { cout<<"Некоректне значення варіанта!"<<endl;
            system("pause>>void");
            return 0;
          }
}
cout<< endl << "Результати:" << endl;
if (x<a) { y=sin(exp(fabs(a + c)))+pow(x+b,2); k=1;}
if (x>=a && x<b) { y=pow(a+fabs(5.3*b),1.0/3)+c; k=2;}
if (x>=b) { y=pow(cos(a+b),2)+sin(x*x); k=3;}
cout<<"y= " << y << " була виконана умова №" << k << endl;
system("pause>>void");
return 0;
}

```

Результати роботи:



```

Введіть номер варіанта (ціле значення) 1, 2 або 3: 2
Введіть довільне значення x= 12.5

Результати:
y= -0.737315 була виконана умова №3

Введіть номер варіанта (ціле значення) 1, 2 або 3: 7
Введіть довільне значення x= 0.78
Некоректне значення варіанта!

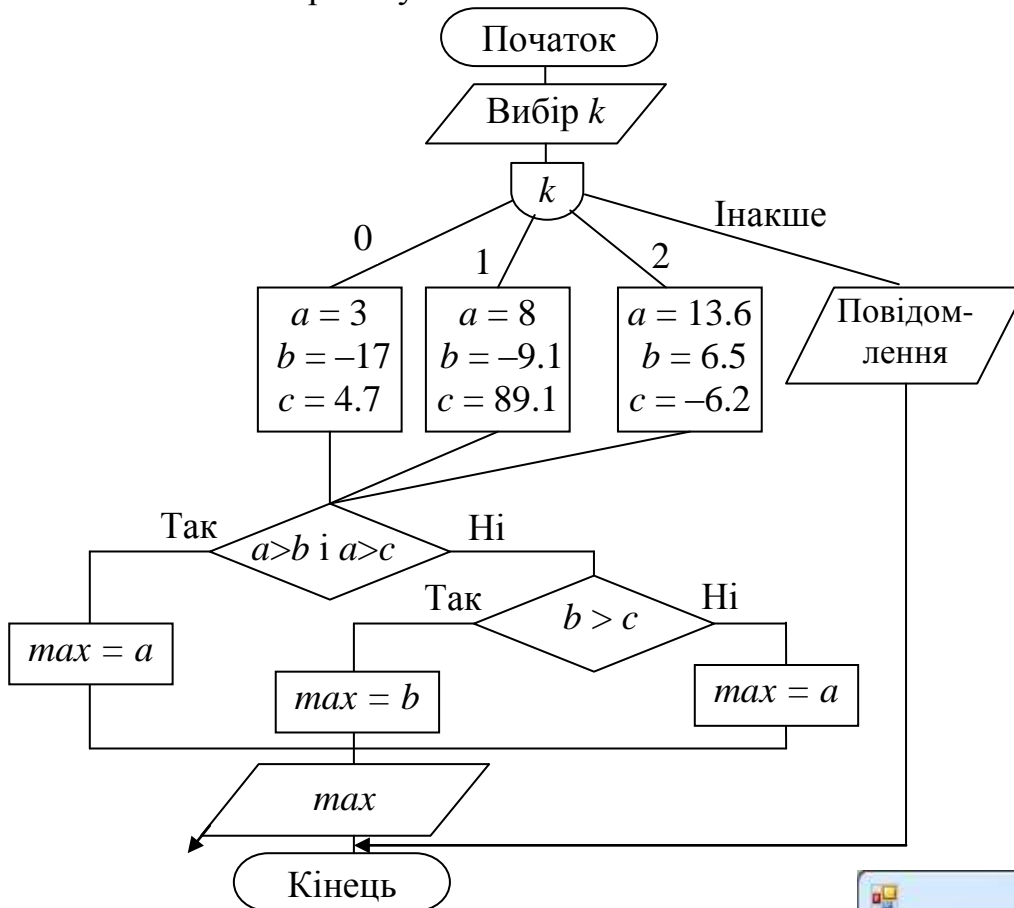
```

Приклад 3. Для будь-якого варіанта з трьох чисел визначити і вивести найбільше з них:

- 1) $a = 3$; $b = 17$; $c = -4.7$;
- 2) $a = 8$; $b = -9.1$; $c = 89.1$;
- 3) $a = 13.6$; $b = 6.5$; $c = -6.2$.

Розв'язок. Пошук найбільшого числа зорганізуємо за допомогою умовної операції `?:`, а вибір варіантів чисел – за допомогою елемента-списка `listBox`, для якого у властивості `Items` сформуємо три рядки з наборами даних. Далі в програмному коді буде опитуватись значення номера вибраного елемента списку (властивість `SelectedIndex`), нумерація яких розпочинається з 0.

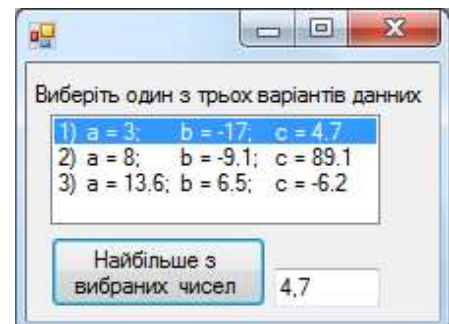
Схема алгоритму:



Текст програми:

```

private: System::Void button1_Click(System::Object^
    sender, System::EventArgs^ e)
{
    double a, b, c, max;
    int k = listBox1->SelectedIndex;
    switch(k)
    {
        case 0: a=3; b=-17; c=4.7; break;
        case 1: a=8; b=-9.1; c=89.1; break;
        case 2: a=13.6; b=6.5; c=-6.2; break;
        default: MessageBox::Show("Виберіть один з трьох варіантів даних");
                return;
    }
    max = (a>b && a>c)? a : (b>c)? b : c;
    textBox1->Text = Convert::ToString(max);
}
  
```



Конструкцію з двома вкладеними одна в одну умовними операціями ? :

```
max = (a>b && a>c)? a : (b>c)? b : c;
```

можна замінити двома вкладеними один в один операторами if-else, але така конструкція є більш громіздкою:

```

if(a>b && a>c) max=a;
else if(b>c) max=b;
else max=c;
  
```

Питання для самоконтролю

- 1) Які оператори в C++ використовуються для організації розгалужень?
- 2) Вкажіть значення змінної `f` після виконання операторів:

```
int f = 1, n = 3, i = 2;
M1: if (i > n) goto PP;
    f = f * i; i++; goto M1;
PP: ;
```

- 3) Вкажіть значення `y` після виконання фрагментів програми:

а) `double y=0; int n=1;`
`switch (n)`
`{case 1: y=n/4.; break;`
`case 2: y=n*n; break;`
`case 3: y=n; break;`
`}`

б) `double y=0; int n=3;`
`switch (n)`
`{case 1: y=n/4.;`
`case 3: y=n*n;`
`case 5: y=n+1;`
`}`

в) `double y=0; int n=4;`
`switch (n)`
`{case 2: { y=n/4.; break; }`
`case 5: { y=n*n; break; }`
`case 9: { y=n; break; }`
`}`

г) `double y=0; int n=1;`
`switch (n)`
`{case 1: { y=n/4; }`
`case 3: { y=n*n; }`
`case 5: { y=n+1; }`
`}`

- 4) У наведеному фрагменті програми:

```
int nom = pow(2, 3);
switch (nom)
{ case 2 : y=d;      break;
  case 8 : y=d*exp(x); break;
  case 10 : y=d*x;   break;
}
```

оператор `switch` обчислюватиме вираз ... (запишіть увесь вираз).

Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
- 2) У протоколі лабораторної роботи скласти схеми алгоритмів і написати програми мовою C++ із застосуванням оператора варіантів `switch` для розв'язання завдань, поданих у табл. 4.1 ... 4.3 відповідно до індивідуального варіанта.
- 3) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Таблиця 4.1

Індивідуальні завдання базового рівня складності

№ вар.	Функція	№ вар.	Функція
1	$y = \begin{cases} 2x+1 & \text{за } k=1; \\ \sqrt[3]{1-x^4} & \text{за } k=2; \\ \lg x+5 & \text{за } k=3; \\ \ln\left \frac{1+x}{x^3+\cos x}\right & \text{за } k=4 \end{cases}$	2	$y = \begin{cases} \frac{\operatorname{tg} 1+e^{x+1.2} }{x+\sin x} & \text{за } n=1; \\ \sqrt[4]{\cos \pi+x } & \text{за } n=2; \\ \frac{1+x^{x+1}-\lg x}{x^3+\ln x } & \text{за } n=3 \end{cases}$
3	$y = \begin{cases} \sin e^{x+1.2} & \text{за } n=1; \\ \sqrt[5]{\lg 1+x }, & \text{за } n=2; \\ \operatorname{tg} \cos x + 5\pi/4 & \text{за } n=3; \\ \frac{1+x^{x+1}-x}{x^3+\ln x } & \text{за } n=4 \end{cases}$	4	$y = \begin{cases} 7.8x^3 - \operatorname{tg}(3.1x^2 + 4x) & \text{за } k=1; \\ e^{0.85\sqrt{x}}(x^2+3) & \text{за } k=2; \\ \sin(2x+\pi) + e^{4x} & \text{за } k=3; \\ x \frac{\sqrt[3]{x+\cos(\pi/2+x)}}{x^{2x}+0.1\cdot 10^{-3}} & \text{за } k=4 \end{cases}$
5	$y = \begin{cases} \operatorname{arctg}(2x+1)+1 & \text{за } k=1; \\ \sqrt[3]{1+x^4} & \text{за } k=2; \\ \cos\left(\frac{\pi}{2}-x^x\right) + e^{ x+5 } & \text{за } k=3; \\ \lg \frac{1+x}{x^3+\sqrt{ x }} & \text{за } k=4 \end{cases}$	6	$y = \begin{cases} \frac{4x^2t}{2x-3t+2} & \text{за } n=1; \\ 6.2x - \frac{\ln \sqrt{x^2+0.1}}{\sqrt{ 2x-\cos x }} & \text{за } n=2; \\ 8.3t^3 + x - 0.2 & \text{за } n=3 \end{cases}$
7	$y = \begin{cases} 1/x + \operatorname{arctg}^2 x^3 & \text{за } M=1; \\ 2^{x-1} + \sin^2 x + \lg x & \text{за } M=2; \\ \sqrt{ 1+x } - \sqrt[3]{x} & \text{за } M=3 \end{cases}$	8	$y = \begin{cases} 10^{-3} + \sin x^3 & \text{за } z=1; \\ \sqrt{1+x} + \sin^2 x & \text{за } z=2; \\ \lg(1/x + \sqrt{x}) & \text{за } z=3 \end{cases}$
9	$y = \begin{cases} \sqrt[5]{x+1} & \text{за } k=1; \\ \operatorname{tg}(\cos x + \pi/2) & \text{за } k=2; \\ e^{2x^2} + \sqrt{ 1-x } & \text{за } k=3; \\ \sin^2(x^2+3) & \text{за } k=4; \\ \cos 3x^2 & \text{за } k=5 \end{cases}$	10	$y = \begin{cases} 2x^2 + \lg x & \text{за } n=1; \\ \cos^2 x + 2.8\sqrt[3]{x} & \text{за } n=2; \\ \sin^2 \sqrt{ x } & \text{за } n=3; \\ \ln\left \frac{x+1}{4}\right & \text{за } n=4 \end{cases}$
11	$t = \begin{cases} \sqrt{ 2^x - x^2 } + 0.5 & \text{за } k=1; \\ 1 + \operatorname{arctg}(x) & \text{за } k=2; \\ \sqrt[5]{\pi^2 + x^2} & \text{за } k=3; \\ \lg 6.5 - x^4 & \text{за } k=4 \end{cases}$	12	$y = \begin{cases} 2^{x+1} + 1 & \text{за } k=1; \\ \sqrt[3]{e^{x^2} + x^4} & \text{за } k=2; \\ \lg \sin(\pi-x) & \text{за } k=3; \\ \operatorname{tg} \frac{1+x}{x^3+x^x} & \text{за } k=4 \end{cases}$

Продовження табл. 4.1

№ вар.	Функція	№ вар.	Функція
13	$y = \begin{cases} \sqrt{x+1} - \cos^2 x & \text{за } k=1; \\ e^{0.01} + \ln x^2, & \text{за } k=2; \\ \sqrt{x} + \sin^2(x-\pi) & \text{за } k=3; \\ x + \lg x & \text{за } k=4 \end{cases}$	14	$y = \begin{cases} 3,5x - 7,3x^2 \operatorname{ctgx} & \text{за } L=1; \\ 2,8 \ln x + e^{\sqrt{x}} & \text{за } L=2; \\ \sqrt[3]{3,4x} + x^2 \sin x & \text{за } L=3; \\ 1,7^x + \cos^2 x^2 & \text{за } L=4 \end{cases}$
15	$y = \begin{cases} \sqrt{x} + e^x & \text{за } L=1; \\ \ln x + 0,8 & \text{за } L=2; \\ x^2 / \lg^2 x & \text{за } L=3; \\ x \cos^2 x + \sqrt{x} & \text{за } L=4; \\ e^{0,8x} + x & \text{за } L=5 \end{cases}$	16	$y = \begin{cases} \cos^2(x + \pi/2) & \text{за } S=1; \\ \operatorname{ctg}^2 \sqrt{x} + 1/x & \text{за } S=2; \\ 2 \sin x + \ln x & \text{за } S=3; \\ \frac{0,8x^2}{e^x + x^x + x} & \text{за } S=4 \end{cases}$
17	$y = \begin{cases} \sin(e^{1+x} + 1) + x^2 & \text{за } K=1; \\ \sqrt{2x + \sin x } + x & \text{за } K=2; \\ 1/\cos x^2 + x & \text{за } K=3; \\ 2x - \sin^2 x & \text{за } K=4 \end{cases}$	18	$y = \begin{cases} 2,5a + \sin x^2 & \text{за } N=1; \\ \lg^2 x+1 & \text{за } N=2; \\ \operatorname{tg}(x - \pi/4) & \text{за } N=3; \\ 2x + \frac{\sin x}{\sqrt{x}} & \text{за } N=4 \end{cases}$
19	$S = \begin{cases} \frac{\cos(\pi - x^2)}{x+1} & \text{за } n=1; \\ \operatorname{tg}(\pi x^2) & \text{за } n=2; \\ \sqrt[3]{e^{x+1}} - \ln x & \text{за } n=3; \\ x^2 - 2^x & \text{за } n=4 \end{cases}$	20	$y = \begin{cases} \sqrt{x} + e^x & \text{за } L=1; \\ \ln x + 0,8 & \text{за } L=2; \\ x^2 + \sin^2 x & \text{за } L=3; \\ x \cos x^2 + \sqrt{x} & \text{за } L=4; \\ e^{0,8x} + \ln x & \text{за } L=5 \end{cases}$
21	$y = \begin{cases} \sqrt[3]{x+1} & \text{за } k=1; \\ \sin(\cos x + \pi/2) & \text{за } k=2; \\ e^{1+x^2} + \lg \sqrt{ 1-x } & \text{за } k=3; \\ \sin^3(x^2 + \pi) & \text{за } k=4; \\ \arccos(3 - x^2) & \text{за } k=5 \end{cases}$	22	$y = \begin{cases} \sqrt{e^x - 1} & \text{за } L=1; \\ \lg x + 1/x & \text{за } L=2; \\ 2^{x-1} + \arcsin^2 x & \text{за } L=3; \\ x \cos^2 x + \sqrt{x} & \text{за } L=4; \\ \ln \sqrt{ x+0,1 } & \text{за } L=5 \end{cases}$
23	$y = \begin{cases} \frac{\sqrt{x} + e^x}{e^{0,1x} + \lg x } & \text{за } L=1; \\ \ln x + \operatorname{ctg}(\sqrt{ \pi - x }) & \text{за } L=2; \\ x^2 + \sin^2 x & \text{за } L=3; \\ x \cos^2 x + \sqrt{x} & \text{за } L=4 \end{cases}$	24	$y = \begin{cases} \sin(x - \pi/2) & \text{за } S=1; \\ \operatorname{tg}^3 \sqrt{x} + x & \text{за } S=2; \\ \arcsin^2 x + \lg x & \text{за } S=3; \\ \frac{2,4 - x^2}{e^x + x^x} & \text{за } S=4 \end{cases}$

Закінчення табл. 4.1

№ вар.	Функція	№ вар.	Функція
25	$y = \begin{cases} e^{x^2} + 0.8x^2 & \text{за } K = 1; \\ \ln x^2 + \sin^2 x & \text{за } K = 2; \\ \sqrt{ x } + \lg x & \text{за } K = 3; \\ x + \operatorname{tg}^2(x - \pi) & \text{за } K = 4 \end{cases}$	26	$S = \begin{cases} 4 \ln x^2 - e^{ x } & \text{за } K = 1; \\ \operatorname{ctg}(\sqrt{ \pi - x }) & \text{за } K = 2; \\ \sin^2(x + \pi) & \text{за } K = 3; \\ \operatorname{tg}(x + 10^{-3}) & \text{за } K = 4 \end{cases}$
27	$y = \begin{cases} e^{2x} - \sin^2 x & \text{за } S = 1; \\ \cos^2 x + \ln x & \text{за } S = 2; \\ \sin^2 x - \ln x & \text{за } S = 3; \\ x + \sin \sqrt{x} & \text{за } S = 4 \end{cases}$	28	$y = \begin{cases} \cos^2 x & \text{за } S = 1; \\ \sin x^2 + 1/x & \text{за } S = 2; \\ 2 \ln x + e^x & \text{за } S = 3; \\ 8x^2 - \operatorname{arctg} x & \text{за } S = 4 \end{cases}$
29	$y = \begin{cases} 3x^2 + \operatorname{arctg} x & \text{за } L = 1; \\ 0.2 \ln x + e^{\sqrt{ x }} & \text{за } L = 2; \\ \sqrt[3]{4 - x} + x^3 \sin x & \text{за } L = 3; \\ 1/e^{x^2} + \cos^5 x & \text{за } L = 4 \end{cases}$	30	$y = \begin{cases} \sqrt{x^4 + 1} - \cos x & \text{за } k = 1; \\ e^{0.1x} + \ln x^2 & \text{за } k = 2; \\ \sqrt{5x} + \operatorname{ctg}^2(x - \pi) & \text{за } k = 3; \\ x + \lg x & \text{за } k = 4 \end{cases}$

Таблиця 4.2

Індивідуальні завдання середнього рівня складності

№ вар.	Функції	Варіанти параметрів
1	$y = \begin{cases} \frac{(2u+1)^2}{7\pi+x} & \text{за } u+x > -0.5 \\ \cos^2 u - \sin \frac{u}{3} & \text{за } -0.5 \leq u+x \leq 10^{-3} \\ \frac{\lg(u+x) - e^x}{3.5x} & \text{за } u+x > 10^{-3} \end{cases}$	1 $u = \sin x$ 2 $u = \cos x$ 3 $u = \operatorname{tg} x$
2	$y = \begin{cases} abx - \cos^2(zx) & \text{за } x < 3.5a; \\ (a-x)^2 - \ln(z+x) & \text{за } 3.5a \leq x \leq b; \\ \sqrt{bx-a} + zx^2 & \text{за } x > b \end{cases}$	1 $a = 0.4; b = 2.3; z = e^{2x}$ 2 $a = 0.2; b = 0.8; z = e^x$ 3 $a = 0.7; b = 8.1; z = 0.8$
3	$y = \begin{cases} \sin(bm + \cos(nx)) & \text{за } bm > x^2; \\ \cos(bm - \sin x) & \text{за } bm < x^2; \\ \sqrt{e^{ \cos x }} + \sqrt{ bmx } & \text{за } bm = x^2 \end{cases}$	1 $b = -1.6; m = 0.9; n = -1.4$ 2 $b = 4.5; m = -2; n = 2.2$ 3 $b = -4.5; m = 0.5; n = -1.5$

Продовження табл. 4.2

№ вар.	Функції	Варіанти параметрів
4	$y = \begin{cases} a \sin^2 x + b \cos(zx) & \text{за } x < -\ln(a); \\ a^b - \cos^3(a + zx) & \text{за } -\ln(a) \leq x \leq b; \\ \sqrt{2.5a^3 + (b - zx^2)^6} & \text{за } x > b \end{cases}$	1 $a = 0.2; \quad b = 0.5; \quad z = e^{ax}$ 2 $a = 0.15; \quad b = 0.2; \quad z = e^{2ax}$ 3 $a = 0.9; \quad b = 5; \quad z = e^{2.5ax}$
5	$y = \begin{cases} \sin(e^{a+b}) + x^2 & \text{за } e^{a+b} > e^x; \\ \arctg(abc) + \sqrt[3]{x} & \text{за } e^{a+b} = e^x; \\ \cos(\sqrt{ x + abc }) & \text{за } e^{a+b} < e^x \end{cases}$	1 $a = 4.2; \quad b = 5.3; \quad c = 1.5$ 2 $a = -0.35; \quad b = 1.8; \quad c = -1.8$ 3 $a = 2.8; \quad b = -0.6; \quad c = 2.0$
6	$y = \begin{cases} 2.8 \sin^2 ax - bx^3 z & \text{за } x < a; \\ z \cos(ax + b)^2 + \ln(z) & \text{за } a \leq x \leq b^2; \\ e^{2.5ax} + zabx & \text{за } x > b^2 \end{cases}$	1 $a = -5; \quad b = 2.5; \quad z = \ln bx^3 $ 2 $a = 3; \quad b = 5; \quad z = \ln bx $ 3 $a = -10; \quad b = 3; \quad z = \ln bx^2 $
7	$y = \begin{cases} xe^a + e^{ bc } & \text{за } 1 - x^2 = a + c \\ \sin^2 ax + \cos bc & \text{за } 1 - x^2 > a + c \\ \sqrt{ab^4 + \sqrt[5]{cx^2}} & \text{за } 1 - x^2 < a + c \end{cases}$	1 $a = 3.2; \quad b = -0.7; \quad c = 2.2$ 2 $a = 10.5; \quad b = -2.5; \quad c = 5.6$ 3 $a = 5.4; \quad b = 3; \quad c = 2.6$
8	$y = \begin{cases} \ln mx + n & \text{за } x^2 > m + n \\ e^{\cos mx - n } & \text{за } x^2 = m + n \\ \sqrt[3]{k^2 + \cos^2 x} & \text{за } x^2 < m + n \end{cases}$	1 $k = 3.1; \quad m = 5.15; \quad n = -1.15$ 2 $k = 0.78; \quad m = -2.4; \quad n = 4.36$ 3 $k = 1.1; \quad m = 0.8; \quad n = 0.41$
9	$y = \begin{cases} a \sin^2 x + b \cos(zx + a) & \text{за } x < a^3; \\ (a + bx)^2 - \sin(a + zx) & \text{за } a^3 \leq x \leq b; \\ \sqrt{x - (\sin(bx + z))} & \text{за } x > b \end{cases}$	1 $a = 1.2; \quad b = 7.2; \quad z = e^x$ 2 $a = -1.5; \quad b = 3.2; \quad z = e^{2x}$ 3 $a = 1.7; \quad b = 5.5; \quad z = e^3$
10	$y = \begin{cases} \sqrt[3]{b^2 + \sqrt{ x + c }} & \text{за } \lg a < x \\ \cos(x - b - c) & \text{за } \lg a = x \\ \sin(x + a - b) & \text{за } \lg a > x \end{cases}$	1 $a = 0.1; \quad b = 9.8; \quad c = 11.12$ 2 $a = 10; \quad b = 10.05; \quad c = 6.2$ 3 $a = 100; \quad b = 3.03; \quad c = 7.12$
11	$y = \begin{cases} \ln(\lg kx + mn) & \text{за } 3x > m + n \\ \sin(kmx) + \sqrt{ nx } & \text{за } 3x = m + n \\ e^{\cos x} + e^{m+n} & \text{за } 3x < m + n \end{cases}$	1 $k = 4; \quad m = -14.7; \quad n = -0.6$ 2 $k = 3; \quad m = 6.5; \quad n = 3.15$ 3 $k = 5; \quad m = -12; \quad n = 0.45$

Продовження табл. 4.2

№ вар.	Функції	Варіанти параметрів
12	$y = \begin{cases} e^{ax} - 3.5 \cos^2(z + bx) & \text{за } x \leq a; \\ a + \ln a + bx - 2x & \text{за } a < x \leq b^{3.5}; \\ a + \cos^{3.5}(a + bxz) & \text{за } x > b^{3.5} \end{cases}$	1 $a = -1; b = 3.4; z = \operatorname{tg} bx$ 2 $a = -3.2; b = 5.5; z = \operatorname{tg} bx^2$ 3 $a = -5.2; b = 7.2; z = \operatorname{tg} bx^3$
13	$y = \begin{cases} x^2 e^{2k} + \ln rx & \text{за } \cos x = \cos(rs) \\ \sqrt[3]{x^2} + \sqrt{ k + rsx } & \text{за } \cos x > \cos(rs) \\ \operatorname{arctg}(kx + rs) & \text{за } \cos x < \cos(rs) \end{cases}$	1 $k = 1.33; r = 0.85; s = 3.5$ 2 $k = 0.9; r = 3.3; s = 1.2$ 3 $k = 1.57; r = 0.75; s = 2.15$
14	$y = \begin{cases} 2.5b^2 + ax - 4.5 \cos xz & \text{за } x \leq 5a; \\ (a^2 - 5.4x)^3 + \ln(xz) & \text{за } x > b; \\ \sqrt{6.5b^2 + (a - x^3z)} & \text{за } 5a < x \leq b \end{cases}$	1 $a = 0.5; b = 4.5; z = e^{ax}$ 2 $a = 0.5; b = 3.7; z = e^{2ax}$ 3 $a = 0.5; b = 2.7; z = e^{2.5ax}$
15	$y = \begin{cases} a \cos^2 x + b \sin zx & \text{за } x \leq a; \\ \operatorname{tg}(ax + z) + \sin^2 bx & \text{за } a < x \leq 1.5b; \\ \ln(ax - b) + z^2 & \text{за } x > 1.5b \end{cases}$	1 $a = 4.5; b = 8.4; z = \operatorname{tg}(bx)^2$ 2 $a = 8.2; b = 15.2; z = \operatorname{tg}(bx)^2$ 3 $a = 1.7; b = 0.5; z = \operatorname{tg}(bx^2)$
16	$y = \begin{cases} 3.5 \sin^2(bx + z)^3 - e^{3.5a} & \text{за } x \leq a; \\ \ln(a + b^3x) + a & \text{за } a < x \leq b^{2.5}; \\ \cos^2(a^b + xz) + a^2 & \text{за } x > b^{2.5} \end{cases}$	1 $a = 0.1; b = 0.5; z = e^{2.5ax}$ 2 $a = 1.2; b = 2.5; z = e^{2.5ax}$ 3 $a = 2.5; b = 1.2; z = e^{2.5ax}$
17	$y = \begin{cases} a + \sin bx + \cos x^2 & \text{за } x \leq a; \\ \sqrt{a + bx} + \sin zx & \text{за } a < x < \ln b; \\ \ln(a + bx + z) & \text{за } x \geq \ln b \end{cases}$	1 $a = -1.2; b = 0.75; z = \ln \operatorname{tg}(bx) $ 2 $a = 0.4; b = 2.4; z = \ln \operatorname{tg}(bx) $ 3 $a = 1.1; b = 6.1; z = \ln \operatorname{tg}(bx) $
18	$y = \begin{cases} \sqrt{ ax - \cos^2 b^3x + 5.1c^2 } & \text{за } 1 - x^2 = a + c \\ e^{0.04x} + \ln b^5 \cos x & \text{за } 1 - x^2 > a + c \\ \cos^2(b^3x^2) + \ln bx - a^2 & \text{за } 1 - x^2 < a + c \end{cases}$	1 $a = 3.5; b = -0.73; c = 2.5$ 2 $a = 15.4; b = -5.6; c = 3.5$ 3 $a = 5.1; b = 4; c = 2.7$
19	$y = \begin{cases} \frac{(2z + 1)^2}{3.71 - x^2} & \text{за } z > -0.5; \\ \sin^3 z - \sin \frac{z}{3\pi} & \text{за } -0.5 \leq z \leq 10^{-3}; \\ \frac{\operatorname{tg}(z + x) - e^x}{3.5x} & \text{за } z > 10^{-3} \end{cases}$	1 $z = \arcsin x^3$ 2 $z = \arccos^2 x$ 3 $z = \operatorname{tg} x$

Продовження табл. 4.2

№ вар.	Функції	Варіанти параметрів
20	$y = \begin{cases} (3.5a - 7.3bx + \sin(zx))^3 & \text{за } x < -\ln a \\ a^b - \cos^3(a + zx) & \text{за } -\ln a \leq x < b \\ \sqrt{ \operatorname{tg} a - x } - x^2 & \text{за } x \geq b \end{cases}$	1 $a = 6; b = 3.2; z = e^{1.5ax}$ 2 $a = 3; b = 6; z = e^{1.5ax}$ 3 $a = 2.7; b = 1.8; z = e^{1.5ax}$
21	$y = \begin{cases} e^{ax} + f \cos^5 bx & \text{за } x \leq a; \\ a + \cos^2 bx - \ln(fx) & \text{за } a < x \leq b^2; \\ \cos^2(a + bfx) & \text{за } x > b^2 \end{cases}$	1 $a = 0.8; b = 2.4; f = e^{1.5ax}$ 2 $a = 1.2; b = 4.2; f = e^{2ax}$ 3 $a = 3.4; b = 8.1; f = e^{3ax}$
22	$y = \begin{cases} a + bx + \sin^2 zx^{3.5} & \text{за } x < a; \\ a + \ln ab - zx^3 + \ln x & \text{за } a \leq x \leq b^2; \\ \sqrt{ a + \operatorname{ctg}(zx) } + b \sin x & \text{за } x > b^2; \end{cases}$	1 $a = 0.3; b = 0.9; z = \sin x^2$ 2 $a = 4.3; b = 3.15; z = \sin x^3$ 3 $a = 6.5; b = 3.5; z = \sin^2 x$
23	$y = \begin{cases} \ln bzx + za^{2.5} & \text{за } a^3 < x \leq b; \\ ax^2 + bz^a + \sin^2 zx & \text{за } x > b; \\ \cos(ax + b) + \ln zx & \text{за } x \leq a^3 \end{cases}$	1 $a = 1.5; b = 6.4; z = \ln bx^3 + 1.5 $ 2 $a = 1.9; b = 8.6; z = \ln bx^3 + 3 $ 3 $a = 0.6; b = 2.4; z = \ln bx^3 + 1.8 $
24	$y = \begin{cases} xe^x + (z + 7.7abx) & \text{за } x < a; \\ \operatorname{tg}(ax + z) + \cos^2 bx & \text{за } a \leq x \leq b^2; \\ \ln(\sin^2(a + bx + zx^2)) & \text{за } x > b^2 \end{cases}$	1 $a = 8.7; b = 3.7; z = \operatorname{tg}(bx)$ 2 $a = 9.3; b = 3.5; z = \operatorname{tg}(abx)$ 3 $a = 2.1; b = 5.7; z = \operatorname{tg}(b^2x)$
25	$y = \begin{cases} a + z \cos^2(bx)^3 & \text{за } x < a; \\ a + \sin^2 b^2 + \ln(zx) & \text{за } a \leq x \leq b; \\ \sqrt[3]{0.3b + \sqrt{ (a - z^2 - \cos x) }} & \text{за } x > b \end{cases}$	1 $a = 1.5; b = 5.7; z = \ln \operatorname{tg}(bx) $ 2 $a = 3.7; b = 8.4; z = \ln \operatorname{tg}(bx) $ 3 $a = 4.4; b = 5.6; z = \ln \operatorname{tg}(bx) $
26	$y = \begin{cases} a^2x^3 + \sqrt{b^4 + 1.7} & \text{за } x < 0.2; \\ \operatorname{arctg}(2^x - p) & \text{за } x = 0.2; \\ \sqrt[3]{\ln a + 4.3} + x & \text{за } x > 0.2 \end{cases}$	1 $a = 0.5; b = 1.5; p = -4$ 2 $a = -1; b = 0.5; p = -4$ 3 $a = -2; b = 0; p = -4$
27	$y = \begin{cases} c \sin(b^2x) + b \ln(cx + a) & \text{за } x < a; \\ a + \ln(bx) - \sin^2(a + cx) & \text{за } a \leq x < b; \\ \sqrt{ \cos(a + bx) + cx^2 } & \text{за } x \geq b \end{cases}$	1 $a = 2.2; b = 2.4; c = \ln bx $ 2 $a = 1.6; b = 1.7; c = \ln bx $ 3 $a = 1.3; b = 4.2; c = \ln b^2x $

Закінчення табл. 4.2

№ вар.	Функції	Варіанти параметрів
28	$y = \begin{cases} \sin(e^{a+b}) + x^2 & \text{за } a+b > x; \\ \operatorname{arctg}(abc) + \sqrt[3]{x} & \text{за } a+b = x; \\ \operatorname{arcsin}(\cos^2(\sqrt{ x })) & \text{за } a+b < x \end{cases}$	1 $a = 7,2; \quad b = -1,3; \quad c = 2,5$ 2 $a = 1,47; \quad b = 3,81; \quad c = 2,8$ 3 $a = 4,8; \quad b = 10,6; \quad c = 2,7$
29	$y = \begin{cases} \operatorname{ctg}(x^2 e^{3k}) + \ln r+x & \text{за } x = rs; \\ \sqrt[5]{x^2} + \sqrt{ \operatorname{arcsin} k } & \text{за } x > rs; \\ \operatorname{arctg}(kx + \operatorname{tg}(rs)) & \text{за } x < rs \end{cases}$	1 $k = -0,3; \quad r = 0,85; \quad s = 3,5$ 2 $k = 0,9; \quad r = 3,3; \quad s = 1,2$ 3 $k = -0,7; \quad r = 0,75; \quad s = 2,15$
30	$y = \begin{cases} a^3 + \operatorname{arctg}(\sin^3 bx) + \cos^2 x^2 & \text{за } x \leq a; \\ \sqrt{(a+bx)+2} + \sin zx & \text{за } a < x < \ln b; \\ \operatorname{arctg}(a+bx+z) & \text{за } x \geq \ln b \end{cases}$	1 $a = 1,5; \quad b = 5,7; \quad z = \operatorname{tg}(bx)$ 2 $a = 3,7; \quad b = 8,4; \quad z = \operatorname{tg}(bx)$ 3 $a = 4,4; \quad b = 5,6; \quad z = \operatorname{tg}(bx)$

Таблиця 4.3

Індивідуальні завдання високого рівня складності

№ вар.	Завдання	Варіанти параметрів
1	Визначити, чи лежить точка А в області, обмеженій параболою $y = 2 - x^2$ та віссю абсцис. Відповідь вивести у вигляді повідомлення	1 $x = 3,5; \quad y = 7,2$ 2 $x = -0,5; \quad y = 1,2$ 3 $x = 0,72; \quad y = -3,12$
2	З трьох чисел визначити і вивести на екран середнє за значенням з них	1 $a = 3; \quad b = 3,5; \quad c = -2,1$ 2 $a = 2,1; \quad b = -6,55; \quad c = 0,1$ 3 $a = -9; \quad b = -3,7; \quad c = -0,1$
3	Для точок з координатами x та y визначити, чи лежать вони всередині кола з радіусом R , якщо центром кола є початок координат	1 $x = 3; \quad y = -7; \quad R = 5;$ 2 $x = 12; \quad y = 11; \quad R = 16;$ 3 $x = -9; \quad y = 6; \quad R = 11.$
4	Задано значення трьох сторін трикутника – a, b та c . Визначити, чи є цей трикутник прямокутним	1 $a = 3; \quad b = 3,5; \quad c = -2,1$ 2 $a = 2,1; \quad b = -6,55; \quad c = 0,1$ 3 $a = -9; \quad b = -3,7; \quad c = -0,1$
6	Задано значення трьох чисел – A, B, C . Подвоїти ті числа, для яких $A + B + C > 0$, а якщо це не так, – замінити їх на нулі	1 $A = -3; \quad B = 3,5; \quad C = 0,1$ 2 $A = 58; \quad B = 27; \quad C = -87$ 3 $A = -8; \quad B = -35; \quad C = 42$
7	Для координат точок $A(x_0, y_0)$ та $B(x_1, y_1)$ визначити, яка з точок – A чи B – найменш віддалена від початку координат ($O(0,0)$)	1 $x_0 = 2; \quad y_0 = 2; \quad x_1 = -4; \quad y_1 = 0$ 2 $x_0 = 8; \quad y_0 = 9; \quad x_1 = 12; \quad y_1 = 1$ 3 $x_0 = -3; \quad y_0 = 0,9; \quad x_1 = 2; \quad y_1 = 3$
8	Для трикутників зі значеннями сторін – a, b та c визначити, чи є вони рівнобедреними	1 $a = 3; \quad b = 3,5; \quad c = 1,1$ 2 $a = 3; \quad b = 6,55; \quad c = 6,55$ 3 $a = 0,9; \quad b = 0,9; \quad c = 0,9$

Продовження табл. 4.3

№ вар.	Завдання	Варіанти параметрів
9	Для трьох цілих чисел (a, b, c) визначити, чи є вони трійкою Піфагора ($c^2 = a^2 + b^2$)	1 $a = 3; b = 5; c = 4$ 2 $a = 3; b = 8; c = 11$ 3 $a = 13; b = 5; c = 12$
10	Для трьох точок – $A_1(x_1, y_1)$, $A_2(x_2, y_2)$ та $A_3(x_3, y_3)$ – визначити, чи лежать ці точки на одній прямій	1 $x_1 = 2; y_1 = 2; x_2 = 4; y_2 = 0; x_3 = -2; y_3 = 6$ 2 $x_1 = 8; y_1 = 9; x_2 = 4; y_2 = 0; x_3 = 5; y_3 = 1$
11	Перевірити числа A та B і змінити їхній знак на протилежний, якщо вони мають різні знаки, а якщо це не так, – замінити їх на нулі	1 $A = -3; B = 3.5$ 2 $A = 58; B = 27$ 3 $A = -8; B = -35$
12	З трьох чисел визначити і вивести на екран найменше з них	1 $a = 23; b = 17; c = 47$ 2 $a = 9; b = -8.1; c = 9.1$ 3 $a = 36; b = 65; c = 62$
13	Для трьох чисел – x, y, z – визначити середнє арифметичне та вивести на екран ті з чисел, які за модулем є більші за середнє арифметичне	1 $x = 3.2; y = -7; z = 0.5$ 2 $x = 2.3; y = 3; z = 2.5$ 3 $x = 23; y = -34; z = 89.5$
14	Для точки з координатами x та y визначити, чи лежить вона на кривій $f(x) = \begin{cases} x^3 & \text{за } x > 1; \\ \sqrt{2-x^2} & \text{за } x \leq 1. \end{cases}$ Похибка складає $\text{eps} = 10^{-3}$, тобто $ f(x) - y < \text{eps}$	1 $x = -3; y = 27$ 2 $x = -2; y = 56$ 3 $x = -1; y = 1$ 4 $x = -3; y = -27$
15	Для точки з координатами x та y визначити, чи лежить вона в четвертій координатній чверті	1 $x = -2.8; y = 0.7$ 2 $x = 0; y = -9.5$ 3 $x = 2; y = -51$
16	Для трикутника зі сторонами a, b та c , найменша зі сторін якого є стороною квадрата, визначити, площа якої фігури є більша	1 $a = 3; b = 5; c = 4$ 2 $a = 13; b = 8; c = 11$ 3 $a = 10; b = 5; c = 12$
17	З трьох чисел – x, y, z – визначити і вивести на екран ті з цих чисел, які є менші за їхнє середнє арифметичне	1 $x = -2; y = 1.2; z = 9.5$ 2 $x = 0.5; y = 2; z = -0.15$ 3 $x = 0.4; y = 2.2; z = 9.5$
18	З трьох чисел – x, y та z – визначити і вивести на екран ті з цих чисел, які за модулем є більші за число π	1 $x = -7.2; y = 3.14; z = -2.5$ 2 $x = -4; y = -3; z = 9.15$ 3 $x = 3.14; y = -3.4; z = 0.59$
19	Для трьох цілих чисел (довжин сторін трикутника) визначити, чи можна побудувати трикутник з цими сторонами	1 $a = 8; b = 13.5; c = 1.1$ 2 $a = 3; b = 3.56; c = 0.55$ 3 $a = 1.9; b = 0.9; c = 0.9$

Закінчення табл. 4.3

№ вар.	Завдання	Варіанти параметрів
20	Для точок $A(x_0, y_0)$ та $B(x_1, y_1)$ визначити, яка з них – А чи В – є найменш віддалена від початку координат ($O(0,0)$)	1 $x_0 = 3; y_0 = 3; x_1 = -6; y_1 = 0$ 2 $x_0 = 8; y_0 = 9; x_1 = 12; y_1 = 1$ 3 $x_0 = 3; y_0 = 0.9; x_1 = 2; y_1 = 3$
21	З трьох чисел – a, b, c – додатні піднести до квадрата, а від'ємні – залишити без змін	1 $a = 0; b = 1.5; c = -31.1$ 2 $a = 2; b = -1.56; c = 2.55$ 3 $a = -1.9; b = 2.9; c = -2.9$
22	З трьох цілих чисел – a, b, c – знайти і вивести на екран непарні числа	1 $a = 2; b = 9; c = 474$ 2 $a = 3; b = 0; c = 27$ 3 $a = 4; b = 11; c = 30$
23	Для трьох чисел – a, b, c – визначити кількість коренів рівняння $ax^2 + bx + c = 0$	1 $a = 1; b = 8; c = 16$ 2 $a = -8; b = 29.7; c = 0.11$ 3 $a = 2.5; b = 5; c = 3$
24	Для точок з координатами x та y визначити, чи лежать вони за межами кола з радіусом R , якщо центром кола є початок координат	1 $x = 78; y = -71; R = 85$ 2 $x = 2; y = 11; R = 13$ 3 $x = -7; y = 6; R = 11$
25	З трьох цілих чисел – a, b, c – знайти і вивести на екран числа, які діляться на 3 без остачі	1 $a = 2; b = 9; c = 474$ 2 $a = 3; b = 0; c = 27$ 3 $a = 4; b = 10; c = 30$
26	З трьох цілих чисел – a, b, c – знайти і вивести на екран числа, які завершуються числом 5	1 $a = 550; b = 175; c = -251$ 2 $a = 872; b = -56; c = -255$ 3 $a = -1995; b = 259; c = 89$
27	З трьох чисел знайти і вивести на екран середнє за абсолютним значенням з них	1 $a = 3; b = -3.5; c = -2.1$ 2 $a = 2.1; b = -6.55; c = 0.1$ 3 $a = -9; b = -3.7; c = 11.1$
28	Для точки з координатами x та y визначити, чи лежить вона в першій координатній чверті	1 $x = 12.8; y = 0.7$ 2 $x = 0; y = -9.5$ 3 $x = -12; y = -51$
29	З трьох цілих чисел – a, b, c – знайти і вивести на екран парні числа.	1 $a = 2; b = 9; c = 474$ 2 $a = 3; b = 0; c = 27$ 3 $a = 4; b = 10; c = 30$
30	Визначити для трьох варіантів координат точок з координатами x та y квадрант, в якому вони розміщені	1 $x = 0; y = -2.7$ 2 $x = -2.43; y = -2.2$ 3 $x = 0.13; y = 0.74$

Лабораторна робота № 5

Програмування циклів.

Оператор циклу з параметром **for**

Мета роботи: набути практичних навиків організації циклічних обчислень у C++ з використанням оператора циклу з параметром **for**.

Теоретичні відомості

Циклічні алгоритми

Обчислювальний процес називається *циклічним*, якщо він неодноразово повторюється, доки виконується певна задана умова. Блок повторюваних операторів називають *тілом* циклу. Існують три різновиди операторів циклу:

- оператор циклу **for**;
- оператор циклу з передумовою **while**;
- оператор циклу з післяумовою **do-while**.

Оператор циклу з параметром **for**

Синтаксис оператора:

for (<ініціалізація>; <умова>; <модифікації>) <тіло циклу>;

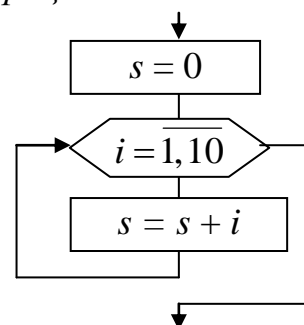
Конструктивно цей оператор складається з трьох основних блоків, розміщених у круглих дужках і відокремлених один від одного крапкою з комою (;), та команд (тіла циклу), які мають багаторазово виконуватись у цьому циклі. На початку виконання оператора циклу одноразово у блоці *ініціалізації* задаються початкові значення змінних (параметрів), які керують циклом. Потім перевіряється *умова* і, якщо вона виконується (**true** або має ненульове значення), то виконується *команда* (чи група команд в операторних дужках {}) тіла циклу. *Модифікації* змінюють параметри циклу і, в разі істинності умови, виконання циклу триває. Якщо *умова* не виконується (**false** або дорівнює нулю), відбувається вихід із циклу, і керування передається на оператор, який іде за оператором **for**. Суттєвим є те, що перевірка умови виконується на початку циклу. Це означає, що тіло циклу може не виконуватись жодного разу, якщо *умова* спочатку є хибна. Кожне повторення (крок) циклу називається *ітерацією*.

Простий приклад для обчислення суми $S = \sum_{i=1}^{10} i$ про-

ілюструє використання оператора **for**:

```
int s = 0;
for(int i = 1; i <= 10; i++) s += i;
```

Цей оператор можна прочитати так: “виконати команду `s += i` 10 разів (для значень `i` від 1 до 10 включно, де `i` на кожній ітерації збільшується на 1)”. У наведеному прикладі є два присвоєння початкових значень: `s=0` і `i=1`, умова продовження циклу: (`i<=10`) і змінення параметра: `i++`. Тілом циклу є команда `s += i`.



Порядок виконання комп'ютером цього циклу є такий:

- 1) присвоюються початкові значення ($s=0, i=1$);
- 2) перевіряється умова ($i \leq 10$);
- 3) якщо умова є істинна (**true**), виконується команда (чи команди) тіла циклу: до суми, обчисленої на попередній ітерації, додається нове число;
- 4) параметр циклу збільшується на 1.

Далі повертаємось до пункту 2. Якщо умову у пункті 2 не буде виконано (**false**), відбудеться вихід із циклу.

В операторі можливі конструкції, коли є відсутній той чи інший блок: *ініціалізація* може бути відсутня, якщо початкове значення задати попередньо; *умова* – якщо припускається, що умова є завжди істинна, тобто слід неодмінно виконувати тіло циклу, доки не зустрінеться оператор **break**; а *модифікації* – якщо приріст параметра здійснювати в тілі циклу або взагалі це є непотрібне. Тоді сам вираз блоку пропускається, але крапка з комою (;) неодмінно має залишитись. Можливою є наявність *порожнього* оператора (оператор є відсутній) у тілі циклу. Наприклад, суму з попереднього прикладу можна обчислити в інший спосіб:

```
for(int s = 0, i = 1; i <= 10; s += i++);
```

У цьому прикладі є відсутній *оператор*, а блок *ініціалізації* вмістить два оператори, розділених операцією “кома”, які задають початкові значення змінних s та i .

Розглянемо використання циклу **for** для обчислення факторіала $F = n!$ (нагадаємо, що факторіал обчислюється за формулою $n! = 1 \cdot 2 \cdot 3 \dots (n-2)(n-1)n$, наприклад: $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$). Наведемо три аналогічні за дією форми запису оператора **for**:

- 1) `int F=1, n=5; for(int i=1; i<=n; i++) F *= i;`
- 2) `int F, i, n=5; for(F=1, i=1; i<=n; F *= i++);`
- 3) `int F=1, i=1, n=5; for(; i<=n;) F *= i++;`

Для завчасного початку чергової ітерації циклу можна використати оператор переходу до наступної ітерації **continue**, наприклад:

```
for(i = 0; i < 20; i++)  
{ if(a[i] == 0) continue;  
  a[i] = 1/a[i]; }
```

Для дострокового виходу з циклу застосовують оператори **break** (вихід з конструкції), **goto** (безумовний перехід) і **return** (вихід з поточної функції).

Розглянемо кілька прикладів розв'язування задач, в яких є доцільне використання оператора циклу **for**. Для усіх програм розроблено алгоритмічні схеми (блок-схеми), які визначають порядок виконання дій, і наведено форми з результатами обчислень.

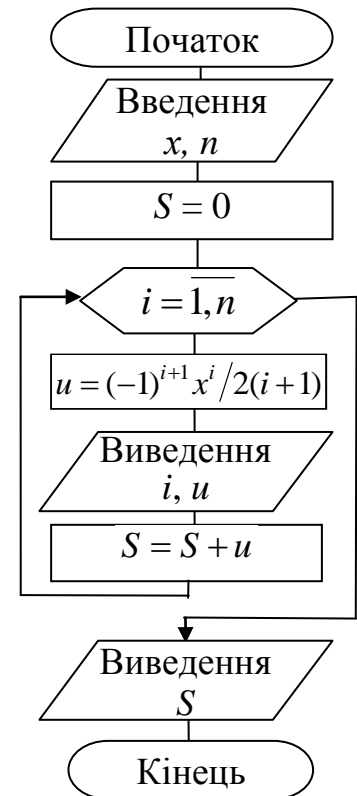
Приклади програм з циклічною структурою, зорганізованих за допомогою оператора for

Приклад 1. Обчислення суми ряду. Ввести ціле число n і дійсне число x , об-

числити
$$s = \frac{x}{4} - \frac{x^3}{6} + \frac{x^5}{8} - \dots = \sum_{i=1}^n \frac{(-1)^{i+1} x^{2i-1}}{2(i+1)}.$$

Текст програми та блок-схема:

```
#include "stdafx.h"
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double x, u, s=0;
    int i, n;
    cout<<"Введіть ціле значення n= ";
    cin>> n;
    cout<<"Введіть значення x= ";
    cin>> x;
    cout<< "\nРезультати:\n";
    for(i=1; i<=n; i++)
    { u = pow(-1,i+1.0)*pow(x,i)/(2*(i+1));
      cout<< "Доданок " << i << " = " << u << endl;
      s+=u;
    }
    cout<< "Сума = " << s << endl;
    system("pause>>void");
    return 0;
}
```



Результати роботи:

```
Введіть ціле значення n= 5
Введіть значення x= 2.4

Результати:
Доданок 1 = 0.6
Доданок 2 = -0.96
Доданок 3 = 1.728
Доданок 4 = -3.31776
Доданок 5 = 6.63552
Сума = 4.68576
```


Приклад 2. Дослідження функцій на певному проміжку (табулювання). Скласти схему алгоритму і програму табулювання функцій (дослідження на певному проміжку) $y(x) = \frac{x^3 \cdot \cos^2 x}{2^x}$ та $z(x) = 2 \sin^3(2x) \ln(0.5 + x)$, змінюючи x на проміжку $[1, 11]$ з кроком $h = 0,2$.

Розв'язок. Для графічного відображення графіків обох функцій доцільно створити програмний проект з формою, на який розмістити відповідні елементи:

- три елементи `textBox` для введення початкового (змінна A), кінцевого (змінна B) значень x та кроку (змінна h), з яким буде змінюватись x . Для створення відповідного пояснювального підпису біля кожного `textBox` розмістити три елементи `label`;

- командну кнопку `button`, при натисненні на яку буде виконуватись програмний код табулювання функцій;

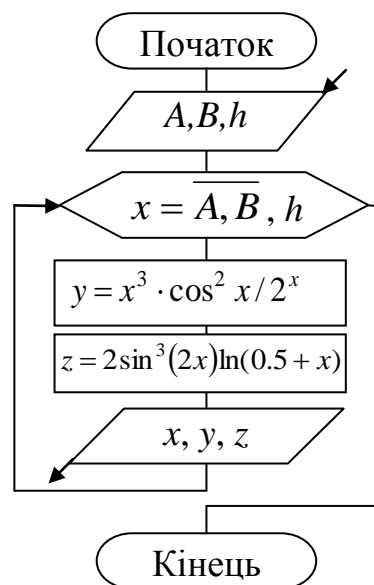
- елемент `richTextBox` для виведення результатів у вигляді трьох стовпців. Встановити для його властивості `Scrollbars` (прокрутки) значення `Both`;

- елемент `Chart` (з групи *Данные*) для виведення діаграми. Налаштувати його, створивши на ньому два графіки (`Series`) з типом `Spline`. Для цього клацнути у властивості `Series` (Коллекция) , що призведе до відкриття діалогового вікна *Редактор коллекции Series*, в якому для першого графіка (`Series1`) вибрати у властивості `ChartType` тип графіка `Spline` (замість значення `Column`). У цьому само вікні знайти ще одну властивість – `LegendText`, щоб записати в ній назву першого графіка: $y = f(x)$. Також можна змінити колір (властивість `Color`), тип лінії (властивість `BorderDashStyle`) та інші параметри графіка.

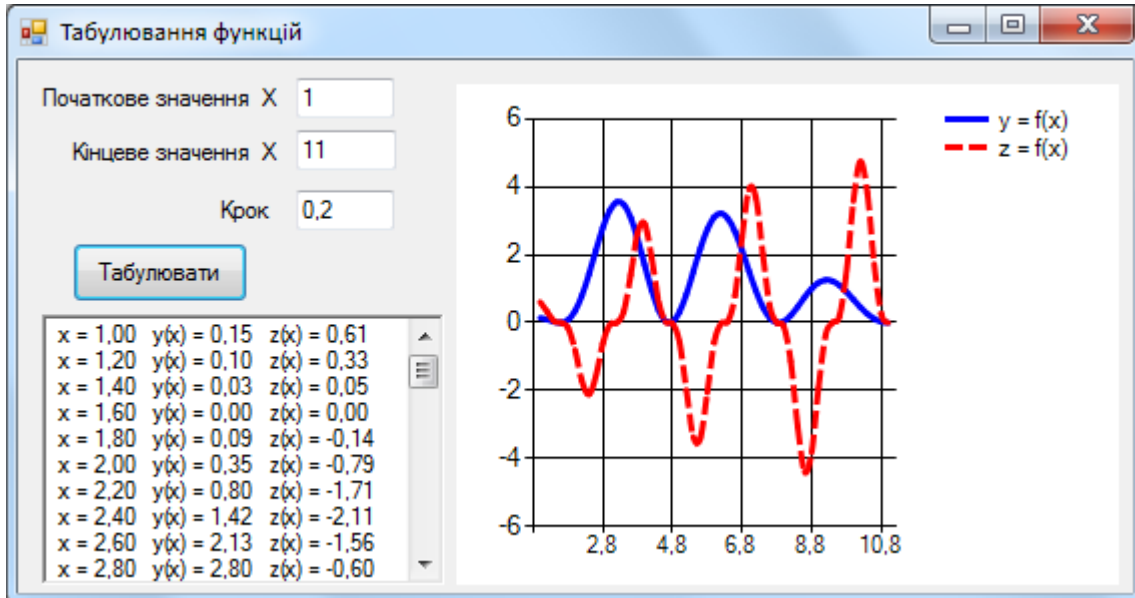
Після цього натиснути кнопку *Добавить*, щоб створити графік `Series2`, та задати для нього такий самий тип і назву другого графіка – $z = f(x)$. Наприкінці натиснути кнопку *OK*, щоб закрити вікно *Редактор коллекции Series*.

Текст програми та блок-схема:

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    richTextBox1->Clear();
    chart1->Series[0]->Points->Clear();
    chart1->Series[1]->Points->Clear();
    double x, y, z, A, B, h;
    A=Convert::ToDouble(textBox1->Text);
    B=Convert::ToDouble(textBox2->Text);
    h=Convert::ToDouble(textBox3->Text);
    for(x=A; x <=B+0.1*h; x+=h)
    {
        y=x*x*x*Math::Pow(Math::Cos(x),2)/Math::Pow(2,x);
        z=2*Math::Pow(Math::Sin(2*x),3)*Math::Log(0.5+x);
        chart1->Series[0]->Points->AddXY(x, y);
        chart1->Series[1]->Points->AddXY(x, z);
        richTextBox1->Text += "x = " + x.ToString("0.00")
            + "    y(x) = " + y.ToString("0.00")
            + "    z(x) = " + z.ToString("0.00") + Environment::NewLine;
    }
}
```



Результати роботи:



Питання для самоконтролю

- 1) Який процес називають циклічним?
- 2) Які оператори циклу використовуються в мові C++?
- 3) Скільки разів виконуватиметься оператор усередині циклу, тобто вкажіть значення s:
`for(int k=-1, s=0; k<=5; k++) s++;`
- 4) Назвіть помилки в таких фрагментах програм:

а) <code>int k, m=2, n=3;</code>	б) <code>int n=-7, m=2;</code>
<code>for(k=1; k<=n; k++)n=n+m;</code>	<code>for(int k=n; k<=m; k--)k++;</code>
- 5) Вкажіть значення m після виконання фрагментів програми:

а) <code>int k, m=1;</code>	б) <code>int m=1, n=5;</code>
<code>for(k=1; k<=5; k++)m++;</code>	<code>for(int k=n; k>=1; k--)m*=k;</code>

Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
- 2) У протоколі лабораторної роботи скласти схему алгоритму і написати програму мовою C++ із застосуванням оператора for для розв'язання завдань, поданих у табл. 5.1 відповідно до індивідуального варіанта (див. приклад програми 1).
- 3) У протоколі лабораторної роботи скласти схему алгоритму і програму табулювання (дослідження) функцій $y = f(x)$ та $z = f(x)$, змінюючи x на заданому проміжку із заданим кроком, які вибрати з табл. 5.2 відповідно до індивідуального варіанта (див. приклад програми 2).
- 4) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Таблиця 5.1

Індивідуальні завдання "Обчислення суми ряду"

№ вар.	Варіанти завдань
1	Ввести натуральне число n і дійсне число x , обчислити $s = 1 + x + \frac{x^2}{2} + \frac{x^3}{3} + \dots = \sum_{i=0}^n \frac{x^i}{i}$
2	Ввести натуральне число n та обчислити $s = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \sum_{i=1}^n \frac{(-1)^{i+1}}{2i-1}$
3	Ввести два натуральні числа n та m ($n < m$), вивести всі кратні 4 числа від n до m та обчислити їхню суму
4	Ввести ціле число n і дійсне x , обчислити $s = \cos(x) + \frac{\cos(2x)}{2} + \frac{\cos(3x)}{3} + \dots = \sum_{i=1}^n \frac{\cos(ix)}{i}$
5	Ввести ціле число n і дійсне число a , обчислити $s = 1 - a + a^2 - a^3 + \dots = \sum_{i=0}^n (-a)^i$
6	Ввести натуральне число n та обчислити $s = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots = \sum_{i=1}^n \frac{1}{i}$
7	Ввести двозначне число N , вивести всі непарні числа від 1 до N та обчислити їхню суму
8	Ввести два натуральні числа n та m ($n < m$), обчислити $s = \sum_{i=n}^m i$
9	Ввести ціле число n і дійсне x , обчислити $s = -\frac{1}{x} + \frac{3}{x^2} - \frac{5}{x^3} + \dots = \sum_{i=1}^n \frac{(-1)^i (2i-1)}{x^i}$
10	Ввести двозначне число N , вивести всі кратні 3 числа від 1 до N та обчислити їхню суму
11	Ввести ціле число n і дійсне число x , обчислити $s = -x + \frac{x^3}{2} - \frac{x^5}{3} + \dots = \sum_{i=1}^n \frac{(-1)^i x^{2i-1}}{i}$
12	Ввести ціле число n і дійсне число x , обчислити $s = \cos(x) + \cos(3x^3) + \dots = \sum_{i=1}^n \cos((2i-1)x^{2i-1})$
13	Ввести ціле число n і дійсне число x , обчислити $s = \frac{1}{2} + \frac{x^2}{5} + \frac{x^3}{8} + \dots = \sum_{i=1}^n \frac{x^{i-1}}{3i-1}$
14	Ввести два натуральні числа n та m ($n < m$), вивести всі парні числа від n до m та обчислити їхню суму
15	Ввести ціле число n і дійсне x , обчислити $s = 1 - \frac{2}{x^2} + \frac{3}{x^4} - \dots = \sum_{i=1}^n \frac{(-1)^{i+1} i}{x^{2i}}$
16	Ввести ціле число n і дійсне x , обчислити $s = \sin(x) + \frac{\sin^2(x)}{4} + \frac{\sin^3(x)}{7} + \dots = \sum_{i=1}^n \frac{\sin^i(x)}{3i-2}$

Закінчення табл. 5.1

№ вар.	Варіанти завдань
17	Ввести ціле число n і дійсне число x , обчислити $s = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots = \sum_{i=0}^n \frac{(-1)^{i-1} x^{2i+1}}{2i+1}$
18	Ввести двозначне число N , вивести всі парні числа від 1 до N та обчислити їхню суму
19	Ввести ціле число n і дійсне число x , обчислити $s = (x+1) + \frac{(x+2)^3}{4} + \frac{(x+3)^5}{9} - \dots = \sum_{i=1}^n \frac{(x+i)^{2i-1}}{i^2}$
20	Ввести двозначне число N , вивести всі кратні 3 числа від 1 до N та обчислити їхню суму
21	Ввести ціле число n і дійсне x , обчислити $s = \sin(1-x) + \frac{\sin(2-x)}{4} + \frac{\sin(3-x)}{9} + \dots = \sum_{i=1}^n \frac{\sin(i-x)}{i^2}$
22	Ввести ціле число n і дійсне число x , обчислити $s = \frac{1}{(x+1)} - \frac{3}{(x+2)^2} + \frac{5}{(x+3)^3} - \dots = \sum_{i=1}^n (-1)^{i+1} \frac{2i-1}{(x+i)^i}$
23	Ввести ціле число n і дійсне число x , обчислити $s = -1 + \frac{x}{2} + \frac{x^2}{7} + \frac{x^3}{14} - \dots = \sum_{i=1}^n \frac{x^{i-1}}{i^2 - 2}$
24	Ввести натуральне число N до 10, вивести всі степені числа 2 від 1 до N та обчислити їхню суму
25	Ввести ціле число n і дійсне x , обчислити $s = \frac{\cos(2x)}{3} + \frac{\cos(4x)}{15} + \frac{\cos(6x)}{35} - \dots = \sum_{i=1}^n \frac{\cos(2ix)}{(2i-1)(2i+1)}$
26	Ввести ціле число n і дійсне x , обчислити $s = \frac{4}{x} + \frac{9}{2x^3} + \frac{16}{3x^5} + \dots = \sum_{i=1}^n \frac{(i+1)^2}{ix^{2i-1}}$
27	Ввести натуральне число n (до 10) і дійсне x , вивести всі степені числа x від 1 до n та обчислити суму цих чисел
28	Ввести ціле число n і дійсне x , обчислити $s = \frac{1}{2} + \frac{\sin(x)}{3} + \frac{\sin^2(x)}{4} + \dots = \sum_{i=0}^n \frac{\sin^i(x)}{i+2}$
29	Ввести ціле число n і дійсне x , обчислити $s = \frac{x}{2} + \frac{x^3}{12} + \frac{x^5}{30} - \dots = \sum_{i=1}^n \frac{x^{2i-1}}{2i(2i-1)}$
30	Ввести ціле число n і дійсне x , обчислити $s = 2 + \frac{3x}{9} + \frac{4x^2}{25} + \frac{5x^3}{49} - \dots = \sum_{i=1}^n \frac{(i+1)x^{i-1}}{(2i-1)^2}$

Індивідуальні завдання
"Дослідження функцій на певному проміжку (табулювання)"

№ вар.	Функція $y = f(x)$	Функція $z = f(x)$	Проміжок дослідження
1	$\sin(x)/x^2$	$\cos(x)/x$	$x \in [0,5; 11], h = 0,3$
2	$\operatorname{arctg}(x + 3,1)$	e^x	$x \in [-6; 1], h = 0,2$
3	$e^{3(x-0.6)}$	$\arcsin(x)$	$x \in [-1; 1], h = 0,05$
4	$\sqrt{ \sin(x + \pi/4) }$	$\sin x^2 + \cos x$	$x \in [-4; 10], h = 0,4$
5	$\operatorname{tg} \sqrt{x}$	$x/(x-3)^2$	$x \in [4,5; 18,5], h = 0,4$
6	$1/e^x$	$\lg(x/2 + 0,1)$	$x \in [0; 7], h = 0,2$
7	$\operatorname{tg}(x/3) \cdot \sin(x - 1,2)$	$2,5 \sin(x/2)$	$x \in [-2; 5], h = 0,2$
8	$1/x$	$(x/3)^2$	$x \in [0,5; 4], h = 0,1$
9	$\cos(1,5x) \cdot \lg(2,5x)$	$e^{\frac{1}{\sqrt{x}}} \sin(x)$	$x \in [3,5; 10,5], h = 0,2$
10	$\cos(x)/x$	$\cos(x/2)$	$x \in [0,3; 7,3], h = 0,2$
11	e^x	$1,5 \cos(x - \pi/4 \cdot e^x)$	$x \in [-6; 1], h = 0,2$
12	$\arcsin(x)$	$\cos(1/(x + \pi/3))$	$x \in [-1; 1], h = 0,05$
13	$\sin^2(x) \cdot \cos(x - \pi)$	$\cos(x)/x$	$x \in [0,5; 11], h = 0,3$
14	$\sin x^2 + \cos x$	$\frac{\sin x}{\lg(x^2 + 2)}$	$x \in [-4; 10], h = 0,4$
15	$x/(x-3)^2$	$ \cos(x/3) $	$x \in [4,5; 18,5], h = 0,4$
16	$\lg(x/2 + 0,1)$	$\cos((x + 2\pi)e^x)$	$x \in [0; 7], h = 0,2$
17	$2,5 \sin(x/2)$	$\sin(x)/\ln(x + 4)$	$x \in [-2; 5], h = 0,2$
18	$(x/3)^2$	$\cos(x + \pi/3) + 1,8$	$x \in [0,5; 4], h = 0,1$
19	$e^{\frac{1}{\sqrt{x}}} \sin(x)$	$\operatorname{tg} \sqrt{x} \cdot \sin\left(x - \frac{\pi}{2}\right)$	$x \in [3,5; 10,5], h = 0,2$
20	$\cos(x/2)$	$\sin(x + \pi/2) \cdot \cos(1/x)$	$x \in [0,3; 7,3], h = 0,2$
21	$\sin^2(x) \cdot \cos(x - \pi)$	$\sin(x)/x^2$	$x \in [0,5; 11], h = 0,3$
22	$1,5 \cos(x - \pi/4 \cdot e^x)$	$\operatorname{arctg}(x + 3,1)$	$x \in [-6; 1], h = 0,2$
23	$\cos(1/(x + \pi/3))$	$e^{3(x-0.6)}$	$x \in [-1; 1], h = 0,05$
24	$\frac{\sin x}{\lg(x^2 + 2)}$	$\sqrt{ \sin(x + \pi/2) }$	$x \in [-4; 10], h = 0,4$
25	$ \cos(x/3) $	$\operatorname{tg} \sqrt{x}$	$x \in [4,5; 18,5], h = 0,4$
26	$\cos((x + 2\pi)e^x)$	$1/e^x$	$x \in [0; 7], h = 0,2$
27	$\operatorname{tg}(x/3) \cdot \sin(x - 1,2)$	$\sin(x)/\ln(x + 4)$	$x \in [-2; 5], h = 0,2$
28	$1/x$	$\cos(x + \pi/3) + 1,8$	$x \in [0,5; 4], h = 0,1$
29	$\cos(1,5x) \cdot \lg(2,5x)$	$\operatorname{tg} \sqrt{x} \cdot \sin(x - \pi/2)$	$x \in [3,5; 10,5], h = 0,2$
30	$\cos(x)/x$	$\sin(x + \pi/2) \cdot \cos(1/x)$	$x \in [0,3; 7,3], h = 0,2$

Лабораторна робота № 6

Циклічне опрацювання послідовностей чисел

Мета роботи: набути практичних навиків програмного опрацювання числових послідовностей за допомогою циклів.

Теоретичні відомості

Існує коло задач, в яких необхідно певним чином опрацьовувати задану числову послідовність, причому для обчислення результату досить переглянути послідовність один раз. Наприклад, щоб обчислити середнє арифметичне заданої послідовності чисел, можна підсумовування чисел і підрахунок їхньої кількості поєднати з введенням чисел. Тоді не потрібно буде зберігати всю послідовність в пам'яті комп'ютера (у вигляді масиву), достатньо мати одну скалярну змінну цілого або дійсного типу і по черзі присвоювати їй значення, які вводяться.

Числова послідовність може задаватися зі зазначенням кількості чисел або мати якусь ознаку кінця.

Приклади програм

Приклад 1. Ввести шість дійсних чисел та визначити найбільше з них.

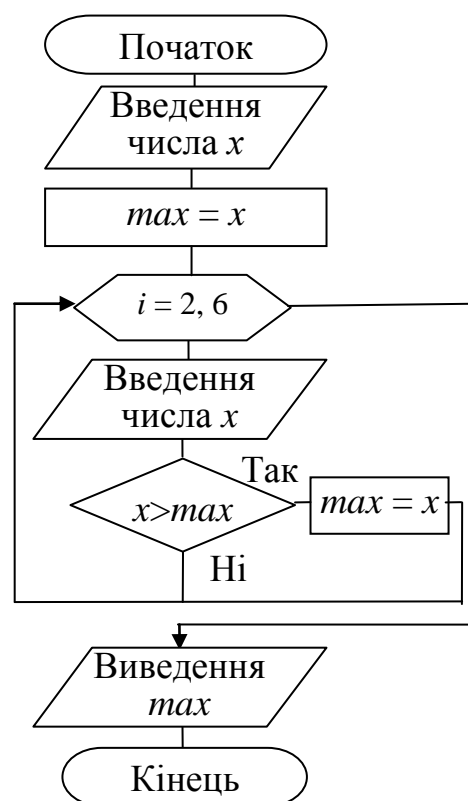
Розв'язок. Алгоритм пошуку максимального числа послідовності:

- 1) ввести перше число x ;
- 2) вважати, що воно є максимальним: $\max = x$;
- 3) у циклі почергово вводити решту чисел. Кожне з введених чисел порівнювати зі значенням \max і, якщо число x буде більшим за \max , запам'ятати його значення як \max :

`if (x>max) max=x;`

Текст програми та блок-схема:

```
#include <iostream>
using namespace std;
int main()
{ setlocale(0, ".1251");
  int i;  double x, max;
  cout << "Ввести 1-ше число: ";
  cin >> x;    max=x;
  for (i=2; i<=6; i++)
  { cout << "Ввести "<< i << " число ";
    cin >> x;
    if (x>max) max=x;
  }
  cout << "Найбільше число: " << max << endl;
  system("pause>>void");
  return 0;
}
```



Результати виконання програми:

Ввести 1 число: 5.5
Ввести 2 число: 0.2
Ввести 3 число: -7
Ввести 4 число: 3.1
Ввести 5 число: 15.2
Ввести 6 число: -1.4
Найбільше число: 15.2

Приклад 2. Ввести 12 цілих чисел та обчислити добуток парних елементів з цієї числової послідовності.

Розв'язок. У циклі, який повторюватиметься 12 разів, виконувати такі дії:

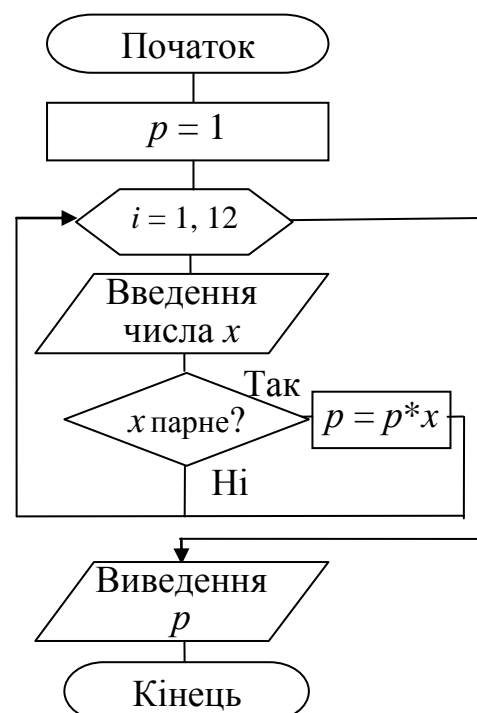
- ✓ виводити запрошення для введення наступного числа;
- ✓ вводити нове число;
- ✓ перевіряти введене число на парність і, якщо це так, то число перемножити на добуток.

Текст програми та блок-схема:

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    int i, x, p=1;                                     // Спочатку добуток дорівнює 1.
    for(i=1; i<=12; i++)                               // У циклі 12 разів повторити дії:
    { cout << "Ввести " << i << " число: ";           // вивести запрошення,
      cin >> x;                                         // ввести число, i,
      if(x%2==0 && x!=0) p *= x;                       // якщо число парне, перемножити його.
    }
    cout << "Добуток парних чисел: " << p << endl;
    system("pause>>void");
    return 0;
}
```

Результати виконання програми:

Ввести 1 число: 3
Ввести 2 число: 1
Ввести 3 число: -7
Ввести 4 число: 6
Ввести 5 число: 3
Ввести 6 число: -2
Ввести 7 число: 4
Ввести 8 число: 7
Ввести 9 число: 11
Ввести 10 число: -4
Ввести 11 число: -5
Ввести 12 число: 3
Добуток парних чисел: 192



У цієї програми є один суттєвий недолік. Якщо парних чисел немає, то результатом обчислення добутку буде виведено 1. Замість цього бажано вивести повідомлення, що парних чисел введено не було. Для цього слід організувати підрахунок кількості введених парних чисел. Якщо після циклу ця кількість становитиме 0, то слід вивести повідомлення. Для обчислення кількості слід оголосити окрему цілу змінну, значення якої спочатку є 0 (парних чисел ще нема). Якщо введене число є парне, кількість збільшиться на 1.

Текст програми:

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    int i, x, p=1, k=0;
    for(i=1; i<=12; i++)
    { cout << "Ввести " << i << " число: ";
      cin >> x;
      if(x%2==0 && x!=0)           // Якщо число парне,
      { p *= x;                     // обчислюватиметься добуток
        k++;                       // і кількість таких чисел.
      }
    }
}
if(k > 0) // Якщо кількість парних чисел є більше за 0, виведеться на екран
    cout << "Добуток парних чисел: " << p << endl; // добуток,
else // інакше – виведеться повідомлення, що парних чисел немає.
    cout << "Парних чисел немає." << endl;
system("pause>>void");
return 0;
}
```

Результати виконання програми:

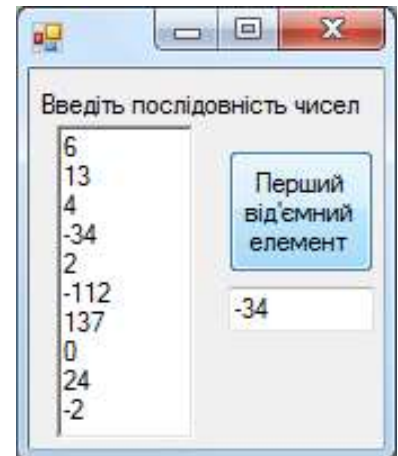
```
Ввести 1 число: 3
Ввести 2 число: 1
Ввести 3 число: -7
Ввести 4 число: 5
Ввести 5 число: 3
Ввести 6 число: -7
Ввести 7 число: 3
Ввести 8 число: 7
Ввести 9 число: 11
Ввести 10 число: -17
Ввести 11 число: -5
Ввести 12 число: 3
Парних чисел немає.
```

Приклад 3. Ввести послідовність цілих чисел і визначити перший від'ємний елемент.

Розв'язок. Для введення та програмного опрацювання довільної кількості чисел доцільно використати елемент `richTextBox1` у режимі створення програмних проектів Windows Forms. У програмі буде опрацьовуватись лише заповнені рядки в `richTextBox1` (кількість рядків – властивість `Lines->Length`).

Текст програми:

```
private: System::Void button1_Click(System::Object^
{
    int v=0, x;
    for(int i=0; i<richTextBox1->Lines->Length; i++)
    { x=Convert::ToInt32(richTextBox1->Lines[i]);
      if (x<0) { v=x; break; }
    }
    textBox1->Text=v.ToString();
}
```

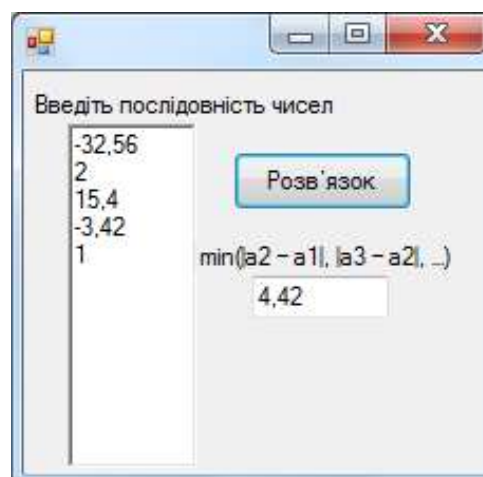
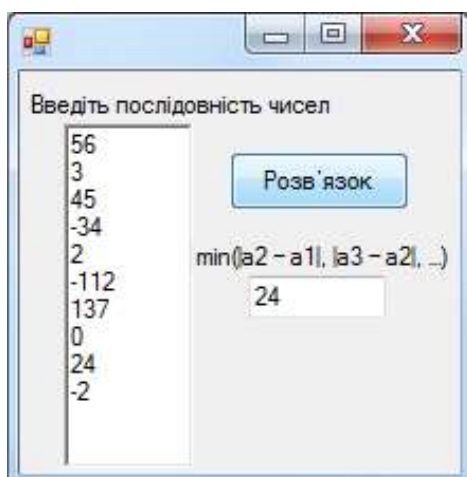


Приклад 4. Ввести послідовність дійсних чисел (a_1, a_2, a_3, \dots) і обчислити $\min(|a_2 - a_1|, |a_3 - a_2|, \dots)$.

Текст програми:

```
private: System::Void button1_Click(System::Object^
{
    double min=1e3, x0, x=Convert::ToDouble(richTextBox1->Lines[0]);
    for(int i=1; i<richTextBox1->Lines->Length; i++)
    { x0=x;
      x=Convert::ToDouble(richTextBox1->Lines[i]);
      if (min>Math::Abs(x-x0)) min=Math::Abs(x-x0);
    }
    textBox1->Text=min.ToString();
}
```

Результати роботи:



Лабораторне завдання

1) У протоколі лабораторної роботи скласти схеми алгоритмів і написати програми мовою C++ із застосуванням оператора циклу для опрацювання числових послідовностей з розв'язання завдань, поданих у табл. 6.1 і 6.2 відповідно до індивідуального варіанта (див. приклади програм 1 ... 4).

2) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Таблиця 6.1

Індивідуальні завдання базового рівня складності

№ вар.	Варіанти завдань
1	Ввести 7 дійсних чисел та обчислити добуток елементів цієї послідовності, значення яких є менше за 6
2	Ввести 10 дійсних чисел та обчислити кількість додатних елементів
3	Ввести 6 дійсних чисел та обчислити суму від'ємних елементів
4	Ввести 5 дійсних чисел і визначити найменше та найбільше серед них
5	Ввести 8 дійсних чисел та обчислити середнє арифметичне ненульових
6	Ввести 9 дійсних чисел та обчислити суму елементів, абсолютне значення яких не перевищує 5
7	Ввести 11 дійсних чисел та обчислити кількість елементів послідовності, значення яких є більше за значення першого елемента
8	Ввести 6 дійсних чисел та обчислити добуток елементів послідовності, значення яких перебувають у діапазоні [3, 6]
9	Ввести 8 дійсних чисел та обчислити середнє арифметичне додатних
10	Ввести 7 дійсних чисел та обчислити суму квадратів тих чисел, модуль яких не перевищує 3
11	Ввести 14 цілих чисел та обчислити кількість ненульових елементів
12	Ввести 9 дійсних чисел та визначити мінімальний елемент послідовності
13	Ввести 6 цілих чисел та обчислити добуток ненульових елементів
14	Ввести 10 цілих чисел та обчислити середнє арифметичне елементів послідовності, значення яких перебувають у діапазоні [10, 20]
15	Ввести 8 дійсних чисел та обчислити кількість елементів, значення яких перебувають у діапазоні [5, 10]
16	Ввести 7 цілих чисел та визначити суму модулів усіх від'ємних елементів
17	Ввести 9 дійсних чисел та обчислити добуток додатних елементів, значення яких не перевищує 4
18	Ввести 12 дійсних чисел та обчислити кількість додатних і кількість від'ємних елементів послідовності
19	Ввести 8 цілих чисел та обчислити середнє арифметичне абсолютних (за модулем) значень усіх елементів послідовності
20	Ввести 6 дійсних чисел та віднайти максимальний і мінімальний елементи та визначити наскільки максимальний елемент є більшим за мінімальний
21	Ввести 11 цілих чисел та обчислити суму тільки двоцифрових елементів

Закінчення табл. 6.1

№ вар.	Варіанти завдань
22	Ввести 9 цілих чисел та обчислити добуток непарних елементів
23	Ввести 14 цілих чисел та обчислити кількість елементів, кратних до числа 3
24	Ввести 7 цілих чисел та обчислити середнє арифметичне парних елементів
25	Ввести 6 дійсних чисел та обчислити суму елементів, значення яких є меншим за значення першого елемента послідовності
26	Ввести 9 цілих чисел та обчислити добуток одноцифрових елементів
27	Ввести 8 цілих чисел та визначити найменший з непарних додатних елементів цієї послідовності
28	Ввести 11 цілих чисел та обчислити середнє арифметичне елементів, кратних до числа 3
29	Ввести 7 цілих чисел та обчислити добуток елементів, кратних до числа 5
30	Ввести 10 цілих чисел та визначити найбільший з парних додатних елементів цієї послідовності

Таблиця 6.2

Індивідуальні завдання середнього рівня складності

№ вар.	Варіанти завдань
1	Ввести послідовність дійсних чисел та обчислити кількість елементів, які більше попереднього елемента послідовності
2	Ввести послідовність дійсних чисел та обчислити суму лише тих елементів цієї послідовності, значення яких є меншими за перший елемент
3	Ввести послідовність дійсних чисел та перевірити, чи є вона упорядкованою за спаданням
4	Ввести послідовність натуральних чисел (a_1, a_2, a_3, \dots) та обчислити $\min(a_1+a_2, a_2+a_3, \dots)$
5	Ввести послідовність цілих чисел та визначити різницю між найменшим і першим числами послідовності
6	Ввести послідовність дійсних чисел (a_1, a_2, a_3, \dots) та обчислити $\min(a_1, a_3, a_5, \dots) + \max(a_2, a_4, a_6, \dots)$
7	Ввести послідовність дійсних чисел (a_1, a_2, a_3, \dots) та обчислити $\max(a_1 - a_2 , a_2 - a_3 , \dots)$
8	Ввести послідовність цілих чисел та визначити різницю між найбільшим і першим числами послідовності
9	Ввести послідовність дійсних чисел (a_1, a_2, a_3, \dots) та обчислити $a_1 * a_2 + a_2 * a_3 + \dots + a_{n-1} * a_n$
10	Ввести послідовність дійсних чисел (a_1, a_2, a_3, \dots) та обчислити $(a_2 - a_1) * (a_3 - a_2) * \dots * (a_n - a_{n-1})$
11	Ввести послідовність дійсних чисел та обчислити середнє арифметичне елементів послідовності, значення яких є меншими за перший елемент

Закінчення табл. 6.2

№ вар.	Варіанти завдань
12	Ввести послідовність цілих чисел та перевірити, чи є в ній однакові сусідні числа
13	Ввести послідовність цілих чисел та з'ясувати, чи складають числа зростаючу послідовність
14	Ввести послідовність цілих чисел та визначити різницю між найбільшим і найменшим числами послідовності
15	Ввести послідовність натуральних чисел та обчислити кількість і суму тих членів послідовності, які діляться на 5 і не діляться на 7
16	Ввести послідовність натуральних чисел та обчислити подвоєну суму всіх додатних членів послідовності
17	Ввести послідовність дійсних чисел та обчислити суму від'ємних і кількість додатних елементів послідовності
18	Ввести послідовність дійсних чисел та віднайти елементи, які за значенням є найближчими, тобто різницями між якими є найменшою
19	Ввести послідовність цілих чисел та обчислити відсотковий вміст від'ємних, нульових та додатних чисел
20	Ввести послідовність цілих чисел та перевірити чи є в ній числа, однакові зі значенням першого елемента цієї послідовності
21	Ввести послідовність натуральних чисел та визначити перший нульовий елемент
22	Ввести послідовність натуральних чисел та обчислити кількість членів послідовності, які мають парні порядкові номери і є непарними числами
23	Ввести послідовність дійсних чисел та обчислити суму квадратів лише тих елементів, значення яких є меншими за перший елемент
24	Ввести послідовність натуральних чисел та обчислити кількість трьохзначних чисел
25	Ввести послідовність цілих чисел та обчислити суму елементів до першого від'ємного значення
26	Ввести послідовність дійсних чисел та обчислити кількість лише тих елементів, значення яких відрізняються від першого елемента на 10
27	Ввести послідовність цілих чисел та обчислити добуток до першого нульового значення
28	Ввести послідовність натуральних чисел та обчислити суму залишків від ділення цих чисел на 2
29	Ввести послідовність дійсних чисел та визначити останній від'ємний елемент
30	Ввести послідовність натуральних чисел та обчислити кількість двозначних чисел

Лабораторна робота № 7

Вкладені цикли

Мета роботи: набути практичних навиків створення програм із вкладеними циклами.

Теоретичні відомості

Цикли може бути вкладено один в одного. При використанні вкладених циклів треба складати програму в такий спосіб, щоб внутрішній цикл повністю вкладався в тіло зовнішнього циклу, тобто цикли не повинні перетинатися. Своєю чергою, внутрішній цикл може містити власні вкладені цикли. Імена параметрів зовнішнього та внутрішнього циклів мають бути різними. Припускаються такі конструкції:

```
for(k=1; k<=10; k++)
{
    . . .
    for(i=1; i<=10; i++)
    {
        . . .
        for(m=1; m<=10; m++)
        {
            ...
        }
    }
}
```

Приклади проектів програм із вкладеними циклами

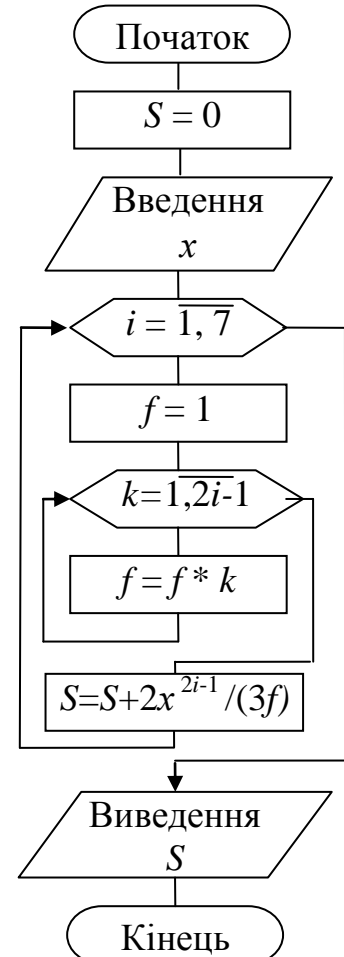
Приклад 1. Обчислити суму ряду $S = \sum_{i=1}^7 \frac{2x^{2i-1}}{3(2i-1)!}$,

де $i = 1, 2, \dots, 7$.

Розв'язок. Для обчислення суми S треба підсумувати сім доданків, для обчислення кожного з яких слід сформувати вкладений цикл для обчислення факторіалів $(2i-1)!$. В наведеній програмі кожний доданок обчислюється в окремій змінній u .

Текст програми та блок-схема:

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double s=0, u, x;
    int i, k, f;
```



```

cout<<"Введіть значення x = ";
cin>>x;
for (i=1; i<=7; i++)
{ f = 1;
  for (k=1; k<=2*i-1; k++) f *= k;
  u = 2*pow(x,2*i-1)/(3*f);
  s += u;
}
cout<<"\nСума = "<<s;
cin.get();
cin.get();
return 0;
}

```

Результати:

Введіть значення x = 2.45

Сума = 3.83416

Приклад 2. Обчислити $S = \sum_{i=1}^m \frac{i}{i-2} \prod_{k=1}^{i+1} \frac{k+3}{k}$, значення m ввести з екрана.

З обчислень вилючити доданки і множники, які дорівнюють нулю в чисельнику або знаменнику.

Розв'язок. В цьому прикладі програми наведені цикли є вкладеними один в одного, оскільки параметр внутрішнього циклу k залежить від параметра зовнішнього циклу i (k змінюється від 1 до $i+1$). Добуток $\prod_{k=1}^{i+1} \frac{k+3}{k}$ є співмножником доданка і обчислюється у внутрішньому циклі у змінній P . Оскільки внутрішній цикл складається лише з одного оператора, то операторні дужки $\{ \}$ не є обов'язковими.

Перед зовнішнім циклом для обчислення суми слід обнулити змінну S , в якій будуть накопичуватись доданки, а перед зовнішнім циклом для обчислення добутку змінній P слід присвоїти значення 1.

Оскільки при обчислюванні добутку беруть участь лише цілі числа, то, щоб при діленні не втратити дробову частину, слід перетворити чисельник до дійсного типу; для цього можна дописати крапку до числа 3: $(k+3.0)/k$.

Перед зовнішнім циклом для обчислення суми слід обнулити змінну S , в якій будуть накопичуватись доданки, а перед зовнішнім циклом для обчислення добутку змінній P слід присвоїти значення 1.

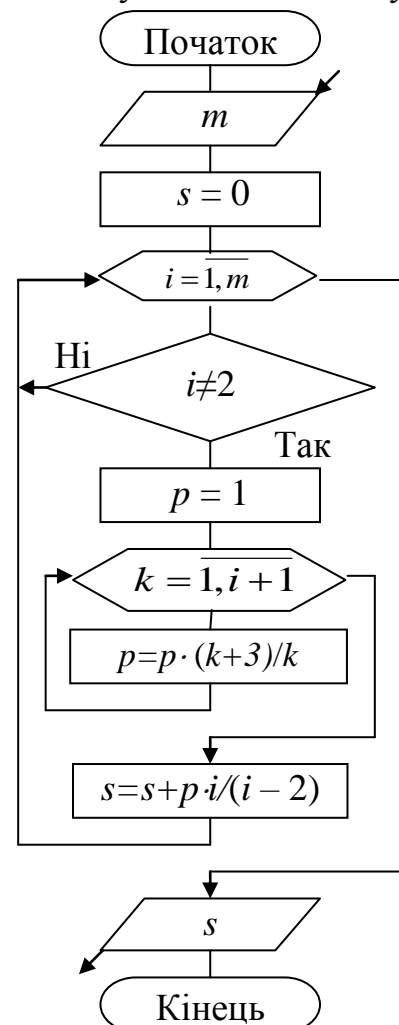
Оскільки при обчислюванні добутку беруть участь лише цілі числа, то, щоб при діленні не втратити дробову частину, слід перетворити чисельник до дійсного типу; для цього можна дописати крапку до числа 3: $(k+3.0)/k$.

Текст програми та блок-схема:

```

#include <iostream>
using namespace std;
int main()

```



```

{
    setlocale(0, ".1251");
    int i, k, m;
    cout<<"Введіть значення m = ";
    cin>>m;
    double S = 0, p;
    for (i = 1; i <= m; i++)
        if (i != 2)
        {
            p = 1;
            for(k = 1; k <= i+1; k++)
                p*=(k+3.)/k;
            S += i/(i-2.) * p;
        }
    cout<<"\nСума = "<<S;
    cin.get();
    cin.get();
    return 0;
}

```

Результати:

Введіть значення m = 10

Сума = 1874.05

Питання та завдання для самоконтролю

- 1) Охарактеризуйте правила організації вкладених циклів.
- 2) Назвіть номер фрагмента програми з вкладеним циклом
 - а) `for(k=1;k<=10;k++)` б) `for(k=1;k<=10;k++)` в) `for(k=1;k<=10;k++)`
`p=k;` `{ p=k;` `{ p=k; }`
`for(j=1;j<=5;j++)` `for(j=1;j<=5;j++)` `for(j=1;j<=5;j++)`
`s+= p*j;` `s+= p*j; }` `{ s+= p*j; }`
- 3) Якого значення набуде змінна s після виконання операторів
 - а) `for(s=0,k=1; k<=3; k++)` б) `for(s=0,k=1; k<=2; k++)`
`for(j=1; j<=k; j++) s+=j;` `for(m=k; m<=2; m++) s+=m;`

Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
- 2) У протоколі лабораторної роботи скласти схеми алгоритмів і написати програми мовою C++ для розв'язання завдань, поданих в табл. 7.1 ... 7.2 відповідно до індивідуального варіанта (див. приклади програм 1 ... 2).
- 3) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Варіанти індивідуальних завдань з теми “Вкладені цикли”

Таблиця 7.1

Індивідуальні завдання базового рівня складності

(довільне значення x слід Ввести)

№	Функція	№	Функція	№	Функція	№	Функція
1	$y = \sum_{k=1}^{10} \frac{x^k}{(k+1)!}$	2	$y = \sum_{i=1}^6 \frac{(-1)^i x^{2i}}{(3i-1)!}$	3	$y = \sum_{i=1}^{10} \frac{(-1)^{i+1} i!}{2^{2i-1} \sin x}$	4	$y = \sum_{i=1}^5 \frac{(2i-1)!}{x^{2i-1}}$
5	$y = \sum_{k=1}^7 \frac{k! \cos(\pi k - x)}{\ln x}$	6	$y = \prod_{k=1}^{10} \frac{x^k}{(2k)!}$	7	$y = \sum_{k=1}^7 \frac{(2k-1)!}{2^k x^{k-1}}$	8	$y = \sum_{i=1}^5 (-1)^{i+1} \frac{\cos x^i}{2i!}$
9	$y = \sum_{i=1}^5 (-1)^{i+1} \frac{\sin x^i}{(2i-1)!}$	10	$y = \sum_{i=1}^3 \frac{x^{2i} (2i-1)!}{2^i}$	11	$y = \sum_{i=1}^6 \frac{(-1)^{i+1} x^{2i}}{(2i-1)!}$	12	$y = \sum_{i=1}^9 \frac{(-1)^i \operatorname{tg} x^2}{(2i+1)!}$
13	$y = \sum_{i=1}^{11} \frac{(-1)^i x^i}{i! \cos\left(i + \frac{\pi}{4}\right)}$	14	$y = \sum_{k=1}^6 \frac{k!}{(1+x)^k}$	15	$y = \prod_{k=1}^5 \frac{(k+1)!}{x^{k+2}}$	16	$y = \sum_{k=1}^5 \frac{x^{k+2}}{k!}$
17	$y = \prod_{k=1}^5 \frac{(-1)^k (2k)!}{4.5 x^{2k-1}}$	18	$y = \sum_{i=1}^6 \frac{\operatorname{tg}(x - \pi)^2}{i!}$	19	$y = \sum_{k=1}^8 \frac{(-1)^k x^{2k-1}}{2^k k!}$	20	$y = \sum_{k=1}^5 \frac{(-1)^k x^{3k-2}}{(k+1)!}$
21	$y = \sum_{i=1}^{11} \frac{(-1)^i x^{3i}}{(2i-1)!}$	22	$y = \sum_{k=1}^8 \frac{2k! \cos(\pi - x)}{x^{2k-1}}$	23	$y = \sum_{k=1}^7 \frac{(-1)^k \cdot k!}{\sqrt{x} + \sin x}$	24	$y = \prod_{k=1}^5 \frac{(-1)^k k!}{x^{k+2}}$
25	$y = \sum_{k=1}^6 \frac{(-1)^k \cdot x^{4k+1}}{(2k)!}$	26	$y = \sum_{k=1}^6 (-1)^k \frac{x^k}{k!}$	27	$y = \sum_{k=1}^5 \frac{(-1)^{k-1} \cdot x^{k+2}}{k!}$	28	$y = \sum_{i=1}^{10} \frac{(2i-1)x^{i+1}}{2i!}$
29	$y = \sum_{i=1}^5 \frac{(-1)^i x^{2i}}{(i+1)! \cos x}$	30	$y = \sum_{i=1}^7 \frac{(i+1)x^i}{i!}$				

Таблиця 7.2

Індивідуальні завдання високого рівня складності

№	Функція	№	Функція	№	Функція
1	$S = \sum_{k=1}^n \frac{(k+1)}{(k-5)} \prod_{m=1}^{k+1} \frac{m-2}{m^2-9}$	2	$Z = \prod_{j=-4}^k \frac{(j+2)}{j-3} \sum_{i=j}^{k+5} \left(\frac{\sqrt[5]{i+5}}{i-11} \right)$	3	$S = \sum_{k=0}^n \frac{(-2)^{k+1}}{(k-5)} \prod_{i=1}^{k+1} \frac{i}{i^2-16}$
4	$Z = \prod_{j=-2}^k \frac{j}{j-1} \sum_{i=j}^k \frac{i}{i+5}$	5	$W = \sum_{i=1}^k \frac{(-1)^i (i+1)!}{i^2-4}$	6	$Y = \sum_{n=1}^k \frac{(-1)^{2-n} (n^2-9)^2}{(n-2)(n+1)!}$
7	$W = \sum_{i=1}^k \frac{(-1)^i}{(i-4)^2} \prod_{n=1}^{i+2} \frac{n^2-4}{n+2}$	8	$L = \prod_{j=1}^k \frac{(j-5)}{j-2} \sum_{i=k}^{12} \frac{\sqrt{i+5}}{i-1}$	9	$Q = \sum_{k=1}^n \frac{(k-1)^{k+1} (k-3)}{(k+1)!}$
10	$Z = \prod_{t=1}^k \frac{k-t-1}{\cos(t)-3} \sum_{i=1}^t \left(\frac{3i-2}{i-7} \right)$	11	$P = \prod_{j=1}^k \frac{(j-6)j}{j-3} \sum_{i=j}^{12} \frac{\sqrt[3]{i+5}}{i-11}$	12	$A = \prod_{j=1}^k \frac{j-3}{(j-4)j} \sum_{i=0}^j \frac{\sqrt{i+4}}{i-1}$
13	$P = \prod_{j=2}^k \frac{(j-6)j}{(j-3)(j-1)!}$	14	$Q = \sum_{k=1}^n \frac{(-1)^k (k-3)^2}{k!}$	15	$U = \prod_{t=1}^k \frac{\sin(t)}{t-3} \sum_{i=1}^t \left(\frac{i+2}{i-7} \right)$

Закінчення табл. 7.2

№	Функція	№	Функція	№	Функція
16	$S = \sum_{k=1}^n \frac{(-3)^{3k-1}}{(k-2)^{3k+1}} \prod_{m=1}^{k+n} \frac{m+3}{m^2-25}$	17	$Z = \prod_{j=-4}^k \frac{j+2}{j(j-3)} \sum_{i=j}^{k+5} \left(\frac{i+5}{i-11} \right)$	18	$Y = \sum_{i=1}^k \frac{(i-4)^i}{(3-i)^2} \prod_{n=i}^{2+k} \frac{n+0.8}{n-i}$
19	$G = \prod_{j=-3}^k \frac{(j-1)^{k+5}}{4j-3} \sum_{i=j}^{k+5} \left(\frac{i+5}{1-i+j} \right)$	20	$D = \sum_{i=-2}^k \frac{(-2^i) \sin^2(i+3)}{(i+3)!}$	21	$R = \sum_{i=0}^k \frac{(1-i)^i}{(i+3)} \prod_{n=i}^{2k} \frac{n-i}{n+2}$
22	$Q = \sum_{k=1}^n \frac{(k-1)^{k+1} (k-7)}{k!}$	23	$W = \sum_{i=1}^k \frac{(-1)^i (i-3)^2}{i!}$	24	$A = \prod_{j=1}^k \frac{(j^2-4)j}{j-k+1} \sum_{i=j}^9 \frac{i-3}{i-7}$
25	$P = \prod_{j=1}^k \frac{j-6}{j-3} \sum_{i=j}^{10} \frac{\sin(i+5)}{i-4}$	26	$Z = \sum_{n=-2}^k \frac{(n+1) n-9 }{(n+3)!}$	27	$p = \sum_{i=1}^n \frac{(2i+1)(i-3)}{(2i-1)!}$
28	$W = \sum_{i=1}^k \frac{(i-1)^i}{(i+3)^2} \prod_{n=1}^{i+1} \frac{n^2-1}{n-2}$	29	$Y = \sum_{i=-1}^k \frac{(k-i)^i (i+2)!}{i^2-4}$	30	$F = \sum_{n=0}^k \frac{(n+2^k) n-4 }{(n)!}$

Лабораторна робота № 8

Оператори циклу **while** та **do-while**

Мета роботи: набути практичних навиків організації циклічних обчислень у C++ з використанням операторів циклу з передумовою **while** та післяумовою **do-while**.

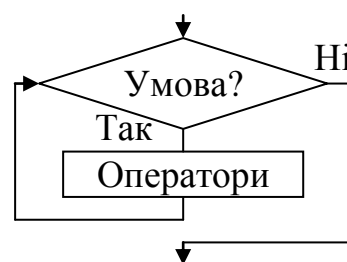
Теоретичні відомості

Оператори з передумовою та післяумовою використовуються для організації циклів і є альтернативними операторові **for**. Звичайно цикл з передумовою використовується, якщо кількість повторювань заздалегідь є невідома або немає явно вираженого кроку змінювання параметра циклу. А для багаторазових повторювань тіла циклу відомим є вираз умови, за істинності якої цикл продовжує виконання. Цю умову слід перевіряти кожного разу перед черговим повторенням.

Синтаксис циклу з передумовою:

```
while (<умова>)  
{ <тіло циклу> };
```

Послідовність операторів (тіло циклу) виконується, доки *умова* є істинна (**true**, має ненульове значення), а вихід з циклу здійснюється, коли *умова* стане хибною (**false**, матиме нульове значення). Якщо *умова* є хибною при входженні до циклу, то *послідовність операторів* не виконуватиметься жодного разу, а керування передаватиметься до наступного оператора програми.

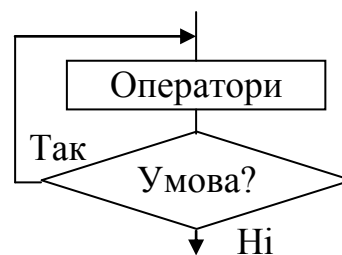


Цикл з післяумовою використовується, якщо є потреба перевіряти умову кожного разу після чергового повторення. Відмінність циклу з передумовою від циклу з післяумовою полягає в першій ітерації: цикл з післяумовою завжди виконується принаймні одноразово незалежно від умови.

Синтаксис циклу з післяумовою:

```
do {  
    <тіло циклу>  
} while (<умова>);
```

Послідовність операторів (тіло циклу) виконується один чи кілька разів, доки *умова* стане хибною (**false** чи дорівнюватиме нулю). Якщо *умова* є істинна (ненульова), то оператори тіла циклу виконуються повторно. Оператор циклу **do-while** використовується в тих випадках, коли є потреба виконати тіло циклу хоча б одноразово, оскільки перевірка умови здійснюється після виконання операторів.



Якщо тіло циклу складається з одного оператора, то операторні дужки **{ }** не є обов'язкові.

Оператори `while` та `do-while` можуть завчасно завершитись при виконанні операторів `break`, `goto`, `return` усередині тіла циклу.

Варто зауважити, що в тілі циклу слід передбачати змінювання параметрів, які беруть участь в умові, інакше умову виходу з циклу ніколи не буде виконано й відбуватиметься зациклювання.

Наведемо відмінності роботи різних операторів циклу на прикладі обчислення суми всіх непарних чисел у діапазоні від 10 до 100:

1) з використанням оператора `for`

```
int i, s=0;
for (i=11; i<100; i += 2) s += i;
```

2) з використанням оператора `while`

```
int s=0, i=11;
while (i<100) { s += i; i += 2; }
```

3) з використанням оператора `do-while`

```
int s=0, i=11;
do { s += i; i += 2;} while (i<100);
```

Приклади програм

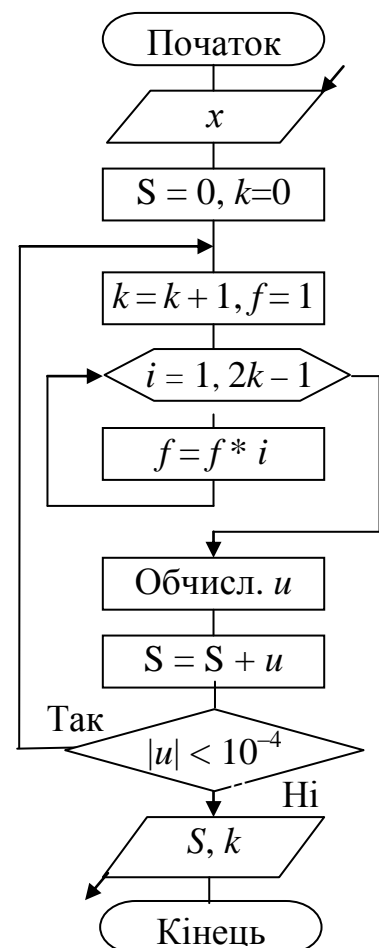
Приклад 1. Обчислити суму ряду $S = \sum_{k=1}^{\infty} \frac{x^{2k+1}}{3^k (2k-1)!}$, підсумовуючи члени

ряду, значення яких за модулем є більше за задану точність $\varepsilon = 10^{-4}$. Визначити кількість доданків. Значення x ($-2 < x < 2$) вводити з клавіатури.

Розв'язок. У цій програмі недоцільно використовувати оператор циклу з параметром `for`, оскільки кількість повторень циклу є наперед невідома. Доцільним буде використання оператора циклу з післяумовою `do-while`, оскільки на момент першої перевірки умови вже треба знати значення першого доданка.

Текст програми та блок-схема:

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double x, f, u, s=0;
    int i, k=0;
    cout << "Введіть значення x= ";
    cin >> x;
    cout << "\nРезультати:\n";
```



```

do                                     // Цикл з післяумовою
{ k++;                                // Збільшення змінної k на 1
  for(i=1,f=1; i<=2*k-1; i++) f *= i; // Обчислення факторіала
  u=pow(x, 2*k+1)/(pow(3.,k)*f);      // Обчислення k-го доданка
  s+=u;                               // Підсумовування доданків
} while(fabs(u)>=1e-4);
cout << "Сума = " << s << endl;
cout << "Кількість доданків = " << k << endl;
system("pause>>void");
return 0;
}

```

Результати роботи:

Введіть значення $x = 2.5$

Результати:

Сума = 7.21478

Кількість доданків = 6

Приклад 2. Обчислити суму ряду $y = \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$, підсумовуючи чле-

ни ряду, значення яких за модулем є більше за задану точність ε . Визначити кількість доданків. Значення x ($-2 < x < 2$) і $\varepsilon = 10^{-4}$ вводити з клавіатури.

Розв'язок. Множник $(-1)^{k+1}$ за непарних $k = 1, 3, 5, \dots$ дорівнює 1, а за парних $k = 2, 4, \dots$ – дорівнює (-1) . Отже, поданий ряд є знакозмінним, де всі парні доданки будуть від'ємними, а всі непарні – зі знаком "+". Для наочності й контролю правильності у розв'язку окремо наведемо всі доданки.

Для того щоби зробити алгоритм програми більш оптимальним, його можна вдосконалити, але для цього слід обчислити рекурентний множник. Це дозволить у даній програмі позбавитись від вкладеного циклу для обчислення факторіала й оператора перевірки на парність k .

```

u = x/8;  s = u;
do
{
  k++;
  cout<< "доданок " << k << " : " << u << endl;
  r = -x*x*(k+2)/((3+k)*(2*k) *(2*k-1));
  u *= r ;
  s += u;
}
while(fabs(u) >= eps);

```

Зупинимось більш докладно на виведенні рекурентної формули.

Подамо суму ряду $y = \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$ у вигляді $y = \sum_{k=1}^{\infty} u_k$, де

$u_k = \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$. Рекурентний множник R – це співвідношення двох поряд розміщених членів ряду:

$$u_2 = R \cdot u_1, \quad u_3 = R \cdot u_2, \quad \dots, \quad u_k = R \cdot u_{k-1},$$

звідки

$$R = \frac{u_k}{u_{k-1}}.$$

Для визначення R в цю формулу слід підставити $u_k = \frac{(-1)^{k+1} x^{2k-1}}{(k+3)(2k)!}$ та u_{k-1} . Для обчислення u_{k-1} підставимо у вираз для u_k $(k-1)$ замість k :

$$\begin{aligned} u_{k-1} &= \frac{(-1)^k x^{2(k-1)-1}}{(k+2)(2k-2)!} = \frac{(-1)^k x^{2k-3}}{(k+2)(2k-2)!}; \\ R = \frac{u_k}{u_{k-1}} &= u_k \cdot \frac{(k+2)(2k-2)!}{(-1)^k x^{2k-3}} = \frac{(-1)^{k+1} \cdot x^{2k-1} \cdot (k+2) \cdot (2k-2)!}{(-1)^k \cdot x^{2k-3} \cdot (k+3) \cdot (2k)!} = \\ &= \frac{(-1)^{k+1-k} \cdot x^{2k-1-(2k-3)} \cdot (k+2)}{(k+3)} \cdot \frac{1 \cdot 2 \cdot \dots \cdot (2k-2)}{1 \cdot 2 \cdot \dots \cdot (2k-2) \cdot (2k-1) \cdot 2k} = -\frac{x^2(k+2)}{2k(k+3)(2k-1)}. \end{aligned}$$

Крім рекурентного множника R , треба обчислити перший член ряду за $k=1$:

$$u_1 = \frac{(-1)^2 x}{4 \cdot (2)!} = \frac{x}{8}.$$

Текст програми:

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    double x, u, r, s=0, eps;
    int i, f, k=0;
    cout<<"Введіть значення x= ";
    cin>> x;
    cout<<"Введіть значення точності ";
    cin>> eps;
    cout<< "\nРезультати:\n";
    u = x/8;  s = u;
    do {
        k++;
        cout<< "доданок " << k << " : " << u << endl;
```

```

    r = -x*x*(k+2)/((3+k)*(2*k) *(2*k-1));
    u *= r ;
    s += u;
} while(fabs(u) >= eps);
cout << "Сума = " << s << endl;
cout << "Кількість доданків = " << k << endl;
system("pause>>void");
return 0;
}

```

Результати роботи:

Введіть значення $x = 2.4$

Введіть значення точності 0.0001

Результати:

доданок 1: 0.3

доданок 2: -0.648

доданок 3: 0.248832

доданок 4: -0.0398131

доданок 5: 0.00351005

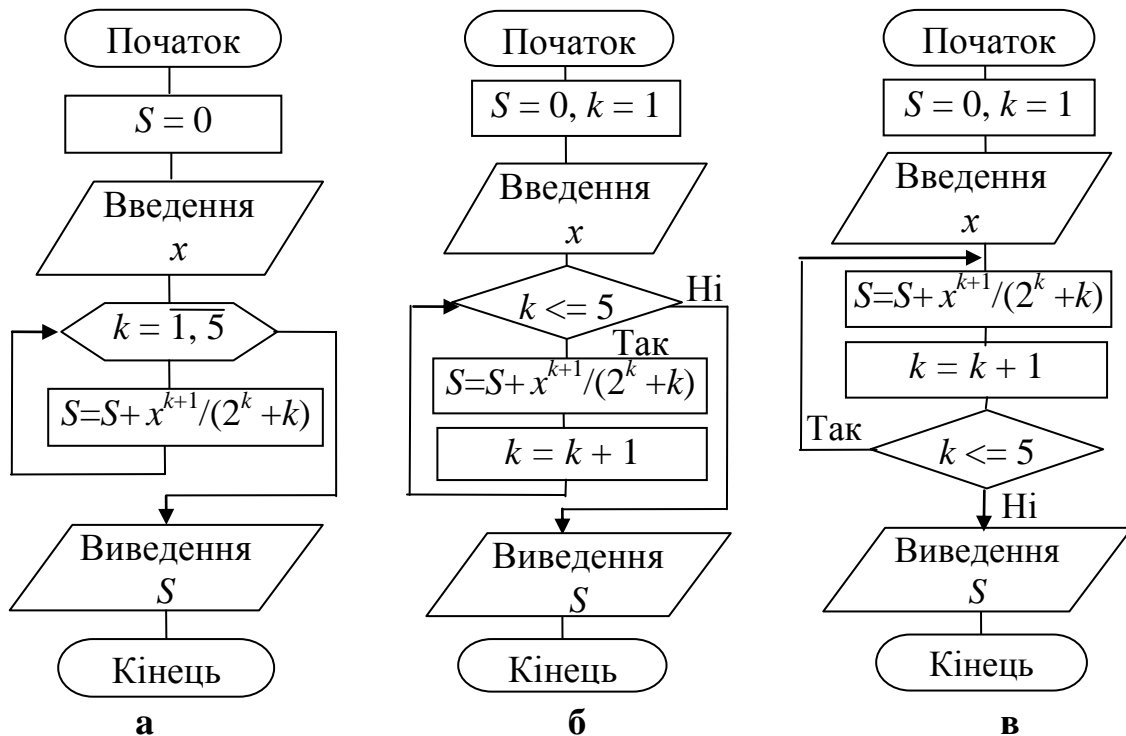
доданок 6: -0.0001196563

Сума = -0.13566

Кількість доданків = 6

Приклад 3. Обчислити суму $f(x) = \sum_{k=1}^7 \frac{x^{k+1}}{2^k + k}$ трьома варіантами, використовуючи різні оператори циклу.

Схеми алгоритмів:



Схеми алгоритмів програм з різними операторами циклу:

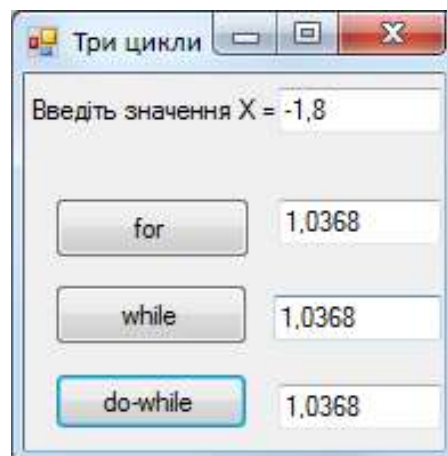
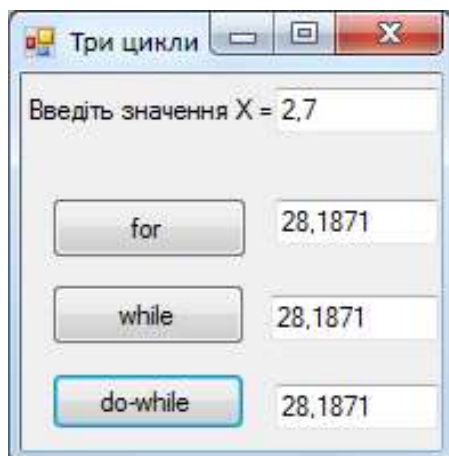
а) **for**; б) **while**; в) **do-while**

Програма з використанням різних операторів циклу:

```
private: System::Void button1_Click(System::Object^ sender,
                                   System::EventArgs^ e)
{
    double S=0, x=Convert::ToDouble(textBox1->Text);
    for(int k=1; k<=5; k++)
        S += Math::Pow(x,k+1)/(Math::Pow(2,k)+k);
    textBox2->Text=S.ToString("0.0000");
}
```

```
private: System::Void button2_Click(System::Object^ sender,
                                   System::EventArgs^ e)
{
    double S=0, x=Convert::ToDouble(textBox1->Text);
    int k=1;
    do {
        S += Math::Pow(x,k+1)/(Math::Pow(2,k)+k);
        k++;
    } while(k<=5);
    textBox3->Text=S.ToString("0.0000");
}
```

```
private: System::Void button3_Click(System::Object^ sender,
                                   System::EventArgs^ e)
{
    double S=0, x=Convert::ToDouble(textBox1->Text);
    int k=1;
    while(k<=5)
    { S += Math::Pow(x,k+1)/(Math::Pow(2,k)+k);
      k++;
    }
    textBox4->Text=S.ToString("0.0000");
}
```



Приклад 4. Визначити і вивести всі дільники введеного натурального числа.

Текст програми:

```
#include <iostream>
int main()
{ unsigned int x, i;
  std::cout << "Enter x: ";
  std::cin >> x;
  std::cout << std::endl;
  for (i=1; i<x; i++)
    if (x%i == 0) // Якщо ділиться без остачі,
                 // вивести черговий дільник
      std::cout << i << "\t";
  std::cin.get();
  std::cin.get();
  return 0;
}
```

Enter x: 75

1 3 5 15 25

Enter x: 122

1 2 61

Приклад 5. Визначити кількість цифр введеного натурального числа.

Текст програми:

```
#include <iostream>
using namespace std;
int main()
{ unsigned int N, kol;
  cout << "N="; cin >> N;
  for (kol=1; N/10>0; kol++,N/=10);
  cout << "kol=" << kol << endl;
  system("pause>>void");
  return 0;
}
```

N=50026
kol=5

Приклад 6. Визначити найменшу та найбільшу цифру введеного натурального числа.

Текст програми:

```
#include <iostream>
using namespace std;
int main()
{ unsigned int N, z, min, max;
  cout<<"N="; cin>>N;
  for (max=min=N%10; N>0; N/=10)
  { z=N%10;
    if (z>max) max=z;
    if (z<min) min=z;
  }
  cout << "min=" << min << endl << "max=" << max << endl;
  system("pause>>void");
  return 0;
}
```

N=550308
min=0
max=8

Приклад 7. Перевірити чи є введене натуральне число паліндромом.

Паліндром – це число (слово чи фраза), яке однаково читається в обох напрямках, або, інакше кажучи, будь-який симетричний відносно своєї середини набір символів. Наприклад, числа 404, 12521 – паліндроми.

Текст програми:

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(0, ".1251");
    unsigned long long n,m,d; // int kol, i; bool Fl;
    cout<<"N="; cin>>n;
    d = 0; m = n;
    while ( m!=0)
    { d = d*10 + m%10;
      m /= 10;
    }
    if (d==n) cout<<"Так, число є паліндромом"<<endl;
    else cout<<"Hi, число не є паліндромом "<<endl;
    system("pause>>void");
    return 0;
}
```

```
N=25052
Так, число є паліндромом
```

Питання для самоконтролю

- 1) Які оператори циклу використовуються у мові C++?
- 2) Назвіть правильну, на Ваш погляд, послідовність номерів, для запису елементів оператора циклу `while`:
 - а) логічний вираз умови;
 - б) оператори тіла циклу;
 - в) `while`.
- 3) Вкажіть значення n після виконання фрагментів програми:
 - а) `int k=0, n=17; while(k<7){ k++; n--; }`
 - б) `int m=1, n=1; while(m<5){ m+=2; n=n*m; }`
- 4) Якими є структура і порядок виконання оператора циклу `do-while`?
- 5) Вкажіть значення y після виконання фрагментів програми:
 - а) `int i=1, y=1; do {y*=i++;} while(y<7);`
 - б) `int k=1; float y=0; do{k+=2;y+=1./k;}while(k<5);`
- 6) Запишіть трьома операторами циклу варіанти обчислення $S = \sum_{i=1}^6 i^2$.
- 7) Вкажіть значення s після виконання операторів


```
s=0.5; i=0;
while(i<5) i++;
s+=1.0/i;
```

Лабораторне завдання

1) У протоколі лабораторної роботи дати відповіді на контрольні питання.

2) У протоколі лабораторної роботи скласти схему алгоритму і програму для обчислення суми нескінченного ряду, підсумовуючи члени ряду, значення яких за модулем перевищують задану точність $\varepsilon = 10^{-4}$. Визначити кількість доданків. Обчислення виконати для x ($-2 < x < 2$), яке ввести. Завдання вибрати з табл. 8.1 відповідно до індивідуального варіанта (див. приклад програм 1 ... 2).

3) У протоколі лабораторної роботи скласти схему алгоритму і написати програму мовою C++ для обчислення функції $f(x)$ трьома варіантами із застосуванням різних операторів циклу (див. приклад 3). Функцію $f(x)$ вибрати з табл. 8.2.

4) У протоколі лабораторної роботи скласти схему алгоритму і написати програму мовою C++ для розв'язання завдання, наведеного у табл. 8.3 (див. приклади програм 4 ... 7).

5) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Таблиця 8.1

Індивідуальні завдання середнього рівня складності

№	Функція $f(x)$	№	Функція $f(x)$	№	Функція $f(x)$
1	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$	2	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{k! 2^k - 1}$	3	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^k}{k!}$
4	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{k(2k+1)!}$	5	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k-1}}{k(k+3)!}$	6	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{2k(k+1)!}$
7	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{(2k-1)!}$	8	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-1}}{(2k)!}$	9	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k+3}}{k^2(k+2)!}$
10	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{(2k+1)!}$	11	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-1}}{(k+2)k!}$	12	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{3k-1}}{(k+3)(3k)!}$
13	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k+1}}{3k(k+1)!}$	14	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2(k+1)}}{(k+2)k!}$	15	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{3(k-2)}}{(k+3)(3k)!}$
16	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{2^k(2k-1)!}$	17	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^k}{(k+4)!}$	18	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{3k-1}}{(k+1)!k^2}$
19	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{2k}}{k! 2^k}$	20	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{3k-1}}{2k(k+3)!}$	21	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k+2}}{k(2k+1)!}$
22	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2(k+1)}}{(2k)!}$	23	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k+1}}{k(2k+1)!}$	24	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-2}}{2k(k+3)!}$
25	$\sum_{k=1}^{\infty} \frac{(-1)^k x^k(2k-1)}{(3k-2)!}$	26	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k-1}}{(2k-1)(k+1)!}$	27	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k-2}}{2^{k+1}k!}$
28	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^{k-1}}{(2+k)(2k+1)!}$	29	$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} x^{3k+1}}{3k(2k-1)!}$	30	$\sum_{k=1}^{\infty} \frac{(-1)^k x^{2k-1}}{2k(2k+1)!}$

Таблиця 8.2

Індивідуальні завдання базового рівня складності

№	Функція $f(x)$	№	Функція $f(x)$	№	Функція $f(x)$
1	$\sum_{k=1}^7 \frac{2^k \sin(x+k)}{(x+1)^k}$	2	$\sum_{k=1}^9 \frac{x^{k+1}}{(k+1)^x}$	3	$\sum_{k=1}^{12} \frac{\sin(kx) + k}{\sqrt[k]{x+0.1+6k}}$
4	$\sum_{k=1}^9 \frac{\ln(x+1)}{(x+k)^k}$	5	$\sum_{k=1}^9 \frac{\sin(2kx) + 0.2}{2k+5}$	6	$\sum_{k=1}^7 \frac{kx \cos(x+k)}{\ln(2+x) + 2k}$
7	$\sum_{k=2}^6 \frac{\sin(0.17x^k)}{2k+x}$	8	$\sum_{k=1}^8 \frac{5 \ln(2kx)}{\arctg(2x) + k^2}$	9	$\sum_{k=2}^9 \frac{\operatorname{tg}(x) - x^2/k}{k^2 - 1}$
10	$\sum_{k=1}^7 \frac{\sin(x^k - \pi)}{\ln k^2 + 0.3}$	11	$\sum_{k=1}^{12} \frac{\cos(kx)}{k}$	12	$\sum_{k=1}^8 \frac{\sin x^k}{4k}$
13	$\sum_{k=1}^7 \frac{\ln^k(3x)}{(2+x)^k}$	14	$\sum_{k=1}^8 \sqrt[k]{\ln(x+1)}$	15	$\sum_{k=6}^1 \frac{x^k}{k^3 + x^{k+2}}$
16	$\sum_{k=1}^{11} \frac{\sin(x^k - 1)}{4k^2 + 1}$	17	$\sum_{k=1}^8 \frac{\ln x^{2k-1}}{2^k(2k-1)}$	18	$\sum_{k=2}^9 \frac{\operatorname{tg}(e^x)}{3k^2 + 1}$
19	$\sum_{k=3}^{10} \frac{x^{k-1} \cos x}{12^k - 1}$	20	$\sum_{k=3}^{11} \frac{\cos^{2+k} x}{2k-1}$	21	$\sum_{k=1}^8 \sin^k(x)(k + \cos(x+2))$
22	$\sum_{k=2}^{10} \frac{\arctg^3(2kx)}{1.2 \ln(k+x)}$	23	$\sum_{k=1}^{11} \frac{\sin(x)^k + 0.3}{(2^k)}$	24	$\sum_{k=1}^6 \frac{k^2 \sin^2(x/k) - kx^2}{e^{kx}}$
25	$\sum_{k=1}^{12} \frac{\cos(x^k)}{(x+5)^k + k}$	26	$\sum_{k=2}^9 \frac{\sin(x+1) + 1.5}{\lg(5kx) + 2.1}$	27	$\sum_{k=1}^7 \frac{2(x+1)^{3-k}}{(k+1)^x + k^3}$
28	$\sum_{k=1}^{10} \cos\left(k^3 - \frac{kx}{5}\right)$	29	$\sum_{k=1}^7 \frac{x \sin(x-k)}{e^{2+x} + k}$	30	$\sum_{k=2}^6 x \cdot \arctg \frac{x - 4.4k}{x + \sin(x+k/5)}$

Таблиця 8.3

Індивідуальні завдання високого рівня складності

№ вар.	Завдання
1	Ввести натуральне число та визначити суму цифр числа
2	Ввести натуральне число та визначити першу цифру числа, наприклад, для числа 3406 цифра 3
3	Ввести натуральне число та визначити суму всіх його дільників
4	Ввести натуральне число та визначити кількість парних цифр числа
5	Ввести натуральне число та визначити найбільшу цифру числа
6	Ввести натуральне число та змінити порядок цифр числа, наприклад, було 1234, стало 4321
7	Ввести натуральне число та визначити його цифри, які кратні 3

Закінчення табл. 8.3

№ вар.	Варіанти завдань
8	Ввести натуральне число та дописати в нього по 1 у початок і кінець
9	Ввести натуральне число та визначити скільки разів його перша цифра зустрічається у числі
10	Ввести натуральне число та визначити кількість його дільників менших 10
11	Ввести натуральне число та визначити чи є воно степенем числа 3
12	Ввести натуральне число та поміняти місцями першу й останню цифру числа, наприклад, з числа 1234 отримати 4231
13	Ввести натуральне число та дописати до нього таке само число, наприклад, з числа 1234 отримати 12341234
14	Ввести натуральне число та визначити кількість його парних дільників
15	Приписати до введенного числа таке само число, наприклад, з числа 1234 здобути 12341234
16	Ввести натуральне число та визначити кількість непарних цифр числа
17	Ввести натуральне число та визначити найменшу цифру числа
18	Ввести натуральне число та визначити його цифри, які кратні 5
19	Ввести натуральне число та визначити чи є воно степенем числа 2
20	Ввести натуральне число та визначити кількість його непарних дільників
21	Вивести всі двозначні числа, які діляться на 5 і містять цифру 5
22	Вивести всі трьохзначні числа, які при діленні на 47 дають в залишку 43, а при діленні на 43 дають в залишку 47
23	Ввести натуральне число та визначити кількість його дільників
24	Ввести натуральне число та визначити суму парних дільників
25	Вивести всі трьохзначні числа, кратні 7, і сума цифр яких також кратна 7
26	Вивести всі двозначні числа, які діляться на 9 або містять цифру 9
27	Вивести всі двозначні числа, сума квадратів цифр яких ділиться на 13
28	Ввести натуральне число та визначити суму непарних дільників
29	Ввести натуральне число та поміняти місцями першу і другу цифру числа, наприклад, з числа 1234 отримати 2134
30	Ввести натуральне число та продублювати кожну з цифр числа двічі, наприклад, з числа 1234 отримати 11223344

Лабораторна робота № 9

Організація функцій в C++

Мета роботи: набути практичних навиків програмної організації функцій засобами C++ для опрацювання різної кількості вхідних і вихідних даних.

Теоретичні відомості

Призначення та поняття функцій

Коли програма стає зовеликою за обсягом і складною для сприймання, є сенс розділити її за змістом на невеликі логічні частини, називані функціями, кожна з яких виконуватиме певне завдання. Унаслідок цього програма стане більш легкою і для розуміння при створюванні, і для процесу налагодження. Окрім того, створення функції позбавляє потреби створювання двох, а то й і більшої кількості, майже однакових фрагментів програмного коду для розв'язання однакових завдань за різних вхідних даних. Розподіл програми на функції є базовим принципом структурного програмування.

Функція – це незалежна іменована частина програми, яка може багаторазово викликатися з інших частин програми, маніпулювати даними та повертати результати. Кожна функція має власне ім'я, за яким здійснюють її виклик. Розрізняють два основні різновиди функцій:

- 1) стандартні вбудовані функції, які є складовою частиною мови програмування, наприклад: `sin()`, `pow()` тощо;
- 2) функції, створювані користувачем для власних потреб.

Створювана у програмі функція повинна бути *оголошеною* і *визначеною*. Оголошення функції має бути написаним у програмі раніш за її використання. Визначення може перебувати у будь-якому місці програми, за винятком тіла (середини) інших функцій.

Оголошення функції складається з *прототипу* (заголовка) і має форму

[клас] <тип результату> <ім'я функції>
(<перелік формальних аргументів із зазначенням типу>);

Наприклад, оголошення функції обчислення середньоарифметичного двох цілих чисел може мати вигляд

`float seredne (int a, int b);`

де `seredne` – ім'я функції;

`a` і `b` – формальні вхідні аргументи (параметри), які за виклику набувають значень фактичних параметрів;

`float` – тип функції, який безпосередньо є типом результату виконання операторів функції. Результат повертається оператором `return`.

Прототип функції може бути розміщено як безпосередньо у програмі, так і в заголовному файлі. У разі, коли визначення функції розміщено у програмі раніш за точку виклику, писати окремо оголошення і окремо реалізацію цієї функції немає потреби й оголошення функції можна уникнути.

Крім оголошення, кожна функція повинна мати визначення (реалізацію).

Визначення (реалізація) функції, крім заголовка, містить тіло функції (команди, які виконує функція) і команду повернення результату `return`:

```
<тип функції> <ім'я функції> (<перелік аргументів>)
{
    <тіло функції>
    return <результат обчислень>;
}
```

Наприклад, визначення функції `seredne` може бути таким:

```
double seredne (int a, int b)
{ return (a+b)/2.0;
}
```

Наведена функція обчислює середньоарифметичне двох цілих чисел `a` та `b` і повертає його значення у точку виклику за допомогою оператора `return`.

Виклик функції `seredne()` може бути в один зі способів:

```
int a=5, x=2, y=-3; double z, s1, s2;
1) z = seredne(5, 4);           // z = 4.5
2) s1 = seredne(a, 11);        // s1 = 8
3) s2 = seredne(x, y);         // s2 = -0.5
```

У першому наведеному виклику числа 5 та 4 є фактичними параметрами, які підставляються у дужки замість формальних параметрів `a` та `b`:

```
double seredne(int a, int b)
                ↑      ↑
z = seredne( 5,    4);
```

У другому виклику першим фактичним параметром є змінна `a`, а другим – число 11. У третьому – обидва фактичних параметри є змінними `x` та `y`.

Якщо функції виконують певні обчислення й дії, які не потребують повернення результатів, за їхній тип вказують тип `void` (тобто порожній, без типу). У таких функціях оператор `return` може бути відсутнім або записуватись без значення, яке повертається. Яскравим прикладом таких функцій є відгуки подій у режимі створення проєктів Windows Forms, наприклад:

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
}
```

Основні правила організації функцій

1) Кількість фактичних параметрів функції має збігатися з кількістю її формальних параметрів.

2) Типи змінних, які є фактичними параметрами, мають збігатися з типами відповідних формальних параметрів, або мати можливість бути перетвореними до типів формальних параметрів.

3) Імена змінних, які є фактичними параметрами, можуть не збігатися з іменами формальних параметрів.

4) Фактичними параметрами можуть бути константи, змінні чи вирази.

5) У прототипі функції необов'язково задавати імена формальних параметрів, достатньо їх оголосити, наприклад у такий спосіб:

```
float seredne(int, int);
```

6) Якщо функція не отримує жодного параметра, слід у заголовку функції ставити порожні дужки (опустити дужки не можна) або записати у дужках `void`:

```
double func();      або      double func(void);
```

7) Тип значення, яке повертає функція, може бути яким-завгодно, але не може бути масивом чи функцією. Наприклад, таке оголошення є помилкове:

```
int [10] func();    // Помилка!
```

8) C++ дозволяє існування у програмі функцій з однаковим ім'ям, але різними параметрами.

Способи передавання параметрів до функції.

Передавання декількох результатів

Існує способи передавання параметрів до функції: *за значенням* і *за адресою*.

При передаванні параметрів (числових змінних) **за значенням** оператори функції працюють з цими числовими значеннями і ніяким чином не можуть повідомити (передати) в головну програму нові значення цих змінних в разі їхнього змінення. За замовчуванням параметри будь-якого типу, крім масиву й функції, передаються до функції за значенням.

При передаванні параметрів **за адресою** передаватимуться не самі значення змінних, а *їхні адреси*. Тобто функція здійснює доступ до комірок пам'яті за цими адресами, а головна програма "бачить" всі змінювання параметрів-змінних так само, як і функція. При цьому існують два різновиди передавання параметрів за адресою: за допомогою *вказівника* (*) і за *посиланням* (&).

При передаванні **вказівника** всередині функції треба використовувати операцію розіменування (розадресації), що робить менш зрозумілим код функції і більш складним налагодження функції. При передаванні **за посиланням** до функції передається посилання, тобто синонім імені, внаслідок чого всередині функції всі звертання до параметра неявно розіменовуються.

Розглянемо приклад функції з передаванням трьох параметрів за значенням, за допомогою вказівника і за посиланням:

```
void f(int i, int* j, int& k)
{ i++; (*j)++; k++;
}
int main()
{ int i=1, j=2, k=3;
  cout << "i  j  k\n";
  cout << i << ' ' << j << ' ' << k << '\n';
  f(i, &j, k);
  cout << i << ' ' << j << ' ' << k;
}
```

Результат роботи цієї програми:

```
i j k
1 2 3
1 3 4
```

Перший параметр *i* передається до функції за значенням, а його змінення у функції не впливає на вихідне значення. Другий параметр *j* передається за

адресою за допомогою вказівника, при цьому для передавання до функції адреси фактичного параметра використовується операція отримання адреси `&j`, а для одержання його значення у функції потрібна операція розіменування (розадресації) `*j`. Третій параметр `k` передається за адресою за допомогою посилання. Це дозволяє передавати фактичний параметр `k` без операції отримання адреси і використовувати його у функції без розадресації.

Якщо функція має повернути понад одне значення, їх можна передати параметрами за посиланням, і ті значення, яких вони набудуть у тілі функції, можна "побачити" і в головній програмі (див. приклад програми 7).

При опрацюванні масивів у функціях їх передавання в якості аргументів завжди здійснюється за адресою, при цьому передається адреса першого елемента (початок масиву), а доступ до кожного з елементів масиву здійснюється як певний зсув (обчислюваний через індекси) від початку масиву. При цьому інформація про кількість елементів втрачається і слід передавати його розмірність через окремий параметр. Докладніше опрацювання масивів у функціях буде розглянуто в подальших лабораторних роботах.

Приклади програм з функціями

Приклад 1. Обчислити значення виразу $x = \frac{\sqrt{6}+6}{2} + \frac{\sqrt{13}+13}{2} + \frac{\sqrt{21}+21}{2}$.

Розв'язок. Усі три доданки заданої формули схожі один на одного, кожен з них можна записати за допомогою спільної формули: $\frac{\sqrt{a}+a}{2}$, де a – число, яке у першому доданку дорівнює 6, у другому – 13, а у третьому – 21. Слушним є створення функції обчислення цієї формули, параметром якої буде a , і тричі викликати цю функцію для кожного з доданків: 6, 13, 21.

Текст функції та її виклику в основній програмі:

```
#include <iostream>
#include <math.h>
using namespace std;

double f (double a)
{ return (sqrt(a)+a)/2;
}

int main ()
{ setlocale(0, ".1251");
  double x = f(6) + f(13) + f(21);
  cout<<" Результат x = "<< x << endl;
  system ("pause>>void");
  return 0;
}
```

Результати роботи консольного додатка:

Результат $x = 25.32$

Приклад 2. Обчислити значення виразу $x = \frac{13 + \sqrt{7}}{7 + \sqrt{13}} + \frac{15 + \sqrt{12}}{12 + \sqrt{15}} + \frac{21 + \sqrt{32}}{32 + \sqrt{21}}$.

Розв'язок. Формула цього прикладу схожа на формулу попереднього завдання і відрізняється лише тим, що функція тут матиме два параметри.

Текст функції та основної програми:

```
include <iostream>
#include <math.h>
using namespace std;
double f(double a, double b)
{ return (a+sqrt(b))/(b+sqrt(a));
}
int main ()
{
    setlocale(0, ".1251");
    double x = f(13,7) + f(15,12) + f(21,32);
    cout<<" Результат x = "<< x << endl;
    system ("pause>>void");
    return 0;
}
```

Результати роботи консольного додатка:

Результат x = 3.37

Приклад 3. Створити функцію обчислення найбільшого спільного дільника (НСД) двох натуральних чисел та визначити за її допомогою НСД для двох і трьох введених чисел.

Розв'язок. Обчислення НСД можна здійснити різними способами.

Спосіб 1. Розкладення чисел на прості множники (див. функцію **nsd1()**) полягає у простому перебиранні всіх чисел з перевіркою на подільність без залишку обох чисел. Перше число, для якого виконається умова перевірки, і буде НСД. Недоліком цього способу є мала швидкість роботи.

Спосіб 2. Існує декілька алгоритмів Евкліда обчислення НСД пари додатних цілих чисел, які використовують різні рекурентні співвідношення. В самому простому випадку **алгоритм Евкліда** формує нову пару: з меншого числа та різниці між більшим і меншим числами. Процес повторюється, доки числа не стануть однаковими. Знайдене число і є найбільшим спільним дільником вихідної пари (див. функцію **nsd2()**).

Спосіб 3. Бінарний алгоритм Евкліда більш швидкий, ніж звичайний алгоритм Евкліда, оскільки замість повільних операцій віднімання виконуються зсуви (див. функції **nsd3()** з оператором циклу **while** та **nsd4()** – з циклом **do-while**).

Спосіб 4. Рекурсивний алгоритм Евкліда (див. функцію **nsd5()**) дозволяє позбавитись від циклічного оператора, який організував чергову ітерацію, за рахунок повторного (рекурсивного) виклику цієї самої функції, але вже зі зменшеними на значення зсуву парою чисел.

Наведемо програмний код для всіх названих способів визначення НСД, а в головній програмі організуємо їх виклик для однієї і тієї самої пари введених чисел.

Текст функції та основної програми:

```
#include <iostream>
using namespace std;

int nsd1( int a, int b)
{
    int n=1;
    if(a > b) n = b; else n = a;
    n++;
    do n--;
    while( a%n != 0 || b%n != 0);
    return n;
}

int nsd2( int a, int b)
{
    while( a!= b)
        if (a > b) a -= b; else b -= a;
    return a;
}

int nsd3( int a, int b)
{
    while ( a && b )
        if ( a>=b ) a %= b;
        else b %= a;
    return a | b;
}

int nsd4( int a, int b)
{
    int n;
    do
    {
        n = a % b;
        a = b;
        b = n;
    } while (b > 0);
    return a;
}

int nsd5( int a, int b)
{
    if(!b) return a; else return nsd5(b, a % b) ;
    // або ще короткий запис: return b ? nsd5 (b, a % b) : a;
}

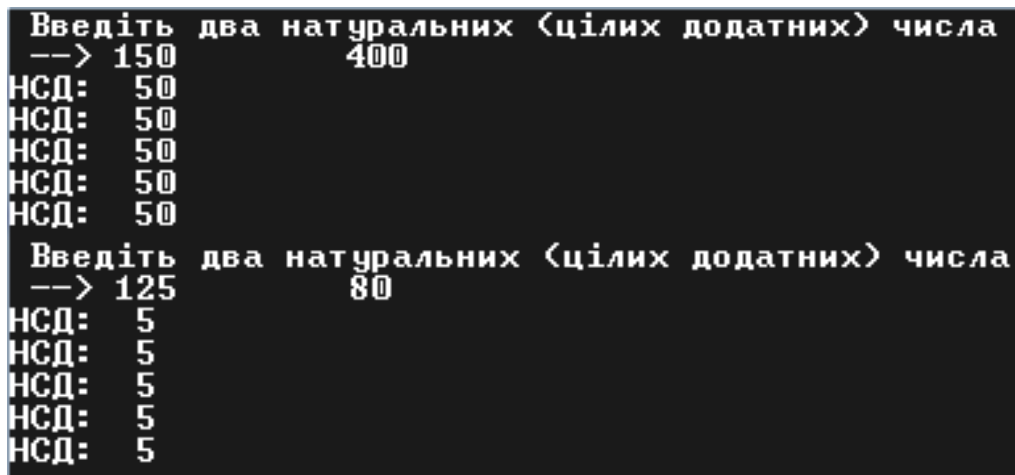
int main()
{
    setlocale(0, ".1251");
```

```

int x, y;
cout << " Введіть два натуральних (цілих додатних) числа \n --> ";
cin >> x >> y;
if (x<=0 || y<=0) cout << "Некоректні дані!" << endl;
else
{ cout<<"НСД: " << nsd1(x,y) <<endl;
  cout<<"НСД: " << nsd2(x,y) <<endl;
  cout<<"НСД: " << nsd3(x,y) <<endl;
  cout<<"НСД: " << nsd4(x,y) <<endl;
  cout<<"НСД: " << nsd5(x,y) <<endl;
}
system ("pause>>void");
return 0;
}

```

Результати роботи консольного додатка:



```

Введіть два натуральних (цілих додатних) числа
--> 150          400
НСД: 50
НСД: 50
НСД: 50
НСД: 50
НСД: 50

Введіть два натуральних (цілих додатних) числа
--> 125          80
НСД: 5
НСД: 5
НСД: 5
НСД: 5
НСД: 5

```

Щоб скористатись будь-якою з наведених функцій для обчислення НСД для трьох введених чисел x, y та z в основній програмі, слід викликати її оператором:

```
cout << "НСД: " << nsd1(x, nsd1(z,y)) <<endl;
```



```

Введіть три натуральних числа
--> 135          99          450
НСД: 9

```

Приклад 4. Створити функцію, яка записуватиме число у зворотному порядку (наприклад, 1208 → 8021), та "перевернути" за її допомогою три введені числа.

Текст функції та основної програми:

```

#include <iostream>
using namespace std;
int reverse(int n)
{ int a=0;
  do
  { a = a*10 + n%10;
    n = n/10 ;
  } while(n);
  return a;
}

```

```

int main()
{ setlocale(0, ".1251");
  int x;
  cout << "Введіть ціле додатне число \n --> ";
  cin >> x ;
  cout<<"Перевернуте число:  "<< reverse(x) <<endl;
  system ("pause>>void");
  return 0;
}

```

Результати роботи консольного додатка:

Введіть ціле додатне число

--> 14089

Перевернуте число: 98041

Приклад 5. Створити функцію, яка переставлятиме місцями першу й останню цифри числа, та за її допомогою змінити таким чином два введені числа.

Текст функції та основної програми:

```

#include <iostream>
using namespace std;

int first_last(int n)
{
  int a = n, i = 1, p = n % 10;
  while (n >= 10)
  {
    i *= 10;
    n /= 10;
  }
  return a - n*i - p + n + p*i;
}

int main()
{
  setlocale(0, ".1251");
  int x;
  cout << " Введіть ціле додатне число \n --> ";
  cin >> x ;
  cout<<"Перевернуте число:  "<< first_last(x) <<endl;
  system ("pause>>void");
  return 0;
}

```

Результати роботи консольного додатка:

Введіть ціле додатне число

--> 120569

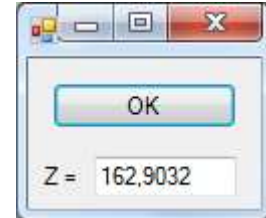
Перевернуте число: 920561

Приклад 6. Обчислити значення виразу $z = \frac{2 \cdot 5! + 3 \cdot 8!}{6! + 4!}$.

Розв'язок. У формулі чотири рази зустрічається факторіал. Тому доречно обчислити значення факторіала у функції і викликати його відповідно чотири рази для різних параметрів.

Текст функції та основної програми:

```
long long fact(int n)
{ long long c=1;
  for(int i=1; i<=n; i++) c*=i;
  return c;
}
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{ double z = (2.0*fact(5)+3*fact(8))/(fact(6)+fact(4));
  textBox1->Text = Convert::ToString(z);
}
```



Приклад 7. Обчислити за допомогою окремої функції значення виразів $y = \arctg^2 z$; $z = \sqrt{x^2 + a}$; $a = \lg \left| \frac{x+b}{e^{b+0.1}} \right|$, значення x та b ввести.

Розв'язок. Обчислити у функції три результуючі значення, одне з яких можна передати через оператор return, а решту – за посиланням. Крім того, слід зважити на порядок обчислення виразів: спочатку обчислити значення a , тоді z (яке залежить від a) і лише згодом – значення y , яке залежить від z .

Текст функції та основної програми:

```
#include <iostream>
#include <math.h>
using namespace std;
double f(double x, double b, double& a, double& z)
{
  a=log10(fabs((x+b)/exp(b+0.1)));
  z=sqrt(x*x+a);
  return pow(atan(z),2);
}
int main()
{
  setlocale(0, ".1251");
  double x, b, a, z;
  cout << " Введіть значення x= "; cin >> x ;
  cout << " Введіть значення b= "; cin >> b ;
  cout<< " y= " << f(x,b,a,z) << endl;
  cout<< " a= " << a << endl << " z= " << z << endl;
  system ("pause>>void");
  return 0;
}
```

```
Введіть значення x= 2.6
Введіть значення b= 0.3
y= 1.46549
a= 0.28868
z= 2.65494
```

Питання та завдання для самоконтролю

- 1) Які дії виконує у функції оператор `return`?
- 2) Який тип має функція, яка не повертає значення?
- 3) Яку інформацію про функцію надає такий її заголовок:
`int fun(double x)`
- 4) В який спосіб можна передавати з функції декілька результатів?
- 5) Чи повинні збігатися імена відповідних формальних і фактичних параметрів функції?

Лабораторне завдання

- 1) У протоколі лабораторної роботи дати відповіді на контрольні питання.
- 2) У протоколі лабораторної роботи скласти схеми алгоритмів і написати програми мовою C++, розв'язання завдань яких організувати у функціях. Тобто основна програма повинна містити введення вхідних даних, виклик функції, в якій реалізовано розв'язання, та виведення результатів. Завдання згідно з індивідуальним варіантом вибрати в табл. 9.1 ... 9.2.
- 3) Створити на комп'ютері програмні проекти у середовищі Visual C++ для реалізації написаних програм. Занести результати обчислень до протоколу.

Таблиця 9.1

Варіанти завдань для створення функції з одним результатом

№ вар.	Завдання
1	Створити функцію обчислення $\frac{x + \sin y}{\sin x + y}$ та визначити $y = \frac{1 + \sin 4}{\sin 1 + 4} + \frac{7 + \sin 5}{\sin 7 + 5} + \frac{3 + \sin 2}{\sin 3 + 2}$ за допомоги цієї функції
2	Створити функцію обчислення факторіала та за її допомоги визначити число різних сполучень із n елементів по m ($n > m$) за формулою $C_n^m = \frac{n!}{m!(n-m)!}$
3	Створити функцію обчислення середнього геометричного трьох чисел за формулою $\sqrt[3]{x \cdot y \cdot z}$ та за допомоги цієї функції визначити для трьох введених чисел значення середнього геометричного
4	Обчислити для двох введених чисел a та b значення $z = \min(a, 2b) + \min(2a - b, b)$, створивши функцію визначення найменшого з двох чисел
5	Для трикутника з довжинами сторін a , b і c значення медіани для сторони a обчислюється за формулою $0.5\sqrt{2b^2 + 2c^2 - a^2}$. Створити функцію для обчислення медіани та за її допомоги визначити медіани для всіх сторін трикутника
6	Створити функцію обчислення $\frac{\sqrt{x} + x}{y + \sqrt{y}}$ та визначити $\frac{\sqrt{5} + 5}{7 + \sqrt{7}} + \frac{\sqrt{12} + 12}{15 + \sqrt{15}} + \frac{\sqrt{22} + 22}{32 + \sqrt{32}}$ за допомоги цієї функції

Продовження табл. 9.1

№ вар.	Завдання
7	Створити функцію для обчислення $\operatorname{sh}(x) = (e^x - e^{-x})/2$ та за її допомоги визначити $\operatorname{sh}(x) \operatorname{tg}(x+1) - \operatorname{ctg}^2(2 + \operatorname{sh}(x-1))$ для введеного значення x
8	Обчислити для двох введених чисел a та b значення $y = \max(2a - b, 2b - a) + \max(a, b)$, створивши функцію визначення найбільшого з двох чисел
9	Створити функцію обчислення суми цифр числа та визначити за її допомоги суми цифр для кожного із трьох введених чисел
10	Створити функцію визначення абсолютного значення (за модулем) числа (без використання стандартних математичних функцій) та обчислити за її допомоги модуль значення для трьох введених чисел
11	Створити функцію обчислення степеня натурального числа з натуральним показником (без використання функції <code>pow</code>) та обчислити за її допомоги значення a^x та x^a для введених натуральних чисел a та x
12	Створити функцію обчислення кількості цифр числа та визначити за її допомоги кількість цифр для кожного із трьох введених чисел
13	Створити функцію обчислення третього степеня значення дійсного числа (без використання функції <code>pow</code>) та визначити для трьох введених чисел значення їх кубів за допомоги цієї функції
14	Створити функцію обміну значеннями двох дійсних змінних та за її допомоги поміняти місцями введені x та y
15	Створити функцію визначення кількості цифр числа та за її допомоги обчислити розмірності трьох введених чисел
16	Створити функцію обчислення факторіала та визначити $z = \frac{3 \cdot 7! - 2 \cdot 6!}{5! + 3!}$ за допомоги цієї функції
17	Визначити, якою кількістю способів можна розсадити 4, 5 та 8 студентів у ряд, якщо число всіх перестановок із n елементів обчислюється як $n!$
18	Скількома можливими способами можна вибрати з 6, 7 та 9 людей делегацію в складі 3 осіб, якщо число всіх сполучень із n елементів по k , де $1 \leq n \leq k$, обчислюється за формулою $C_n^k = \frac{n!}{k!(n-k)!}$
19	Створити функцію обчислення середнього геометричного чотирьох чисел за формулою $\sqrt[4]{a \cdot b \cdot c \cdot d}$ та за допомоги цієї функції визначити для чотирьох введених чисел значення середнього геометричного
20	Створити функцію обчислення $\frac{\sqrt{x} + x}{2}$ та визначити $\frac{\sqrt{5}+5}{2} + \frac{\sqrt{11}+11}{2} + \frac{\sqrt{17}+17}{2}$ за допомоги цієї функції
21	Створити функцію обчислення найбільшого спільного дільника (НСД) двох натуральних чисел за алгоритмом Евкліда та за допомогою неї визначити НСД чотирьох введених чисел

Закінчення табл. 9.1

№ вар.	Завдання
22	Створити функцію обчислення гіпотенузи прямокутного трикутника за значеннями двох катетів та за допомоги цієї функції визначити для двох введених значень катетів значення гіпотенузи
23	Створити функцію обчислення найбільшого спільного дільника (НСД) двох натуральних чисел за алгоритмом розкладення чисел на прості множники та за допомогою неї визначити НСД трьох введених чисел
24	Створити функцію для обчислення $\text{ch}(x) = (e^x + e^{-x})/2$ та за її допомоги визначити $\text{ch}(x) \text{ctg}^2(x+1) - \text{tg}(0.5 - \text{ch}(x+1))$ для введеного значення x
25	Створити функцію обчислення середнього геометричного двох чисел за формулою $\sqrt{x \cdot y}$ та за допомоги цієї функції визначити для двох введених чисел значення середнього геометричного
26	Створити функцію визначення найбільшої цифри числа та за її допомоги обчислити найбільші цифри для трьох введених чисел
27	Створити функцію обчислення значення катета прямокутного трикутника за значенням гіпотенузи та одного з катетів і за допомоги цієї функції визначити значення катета для двох введених чисел
28	Створити функцію обчислення другого степеня дійсного числа (без використання функції pow) та визначити для трьох введених чисел значення їх квадратів за допомоги цієї функції
29	Створити функцію обчислення $\frac{x - \cos y}{\sin x + y}$ і визначити $z = \frac{2 - \cos 3}{\sin 2 + 3} + \frac{11 - \cos 5}{\sin 11 + 5} - \frac{3 - \cos 9}{\sin 3 + 9}$ за допомоги цієї функції
30	Створити функцію обчислення найбільшого спільного дільника (НСД) двох натуральних чисел за бінарним алгоритмом Евкліда та за допомогою неї визначити НСД п'яти введених чисел

Таблиця 9.2

Варіанти завдань для створення функції з трьома результатами

№ вар.	Обчислювані значення	Вхідні параметри
1	$y = a \sin^2 b + b \cos^2 a; a = \sqrt[3]{ b+c }; b = \sqrt{x}$	x, c
2	$y = a^2 + b^2; a = \ln x ; b = e^k + a$	x, k
3	$y = e^x + 5.8^c; c = a^2 + \sqrt{b}; a = b^3 + \ln b $	x, b
4	$y = \sqrt[3]{ a-b }; a = \lg x; b = \sqrt{x^2 + t^2}$	x, t
5	$y = a^3 / b^2; a = e^{\sqrt{ x }}; b = (\sin p^2 + x^3)$	x, p
6	$y = p^2 + t^4; p = x^2 - \sqrt{ x }; t = \sqrt[3]{x + a^2}$	x, a

Закінчення табл. 9.2

№ вар.	Обчислювані значення	Вхідні параметри
7	$y = c^3 / \cos c; c = a^2 + b^2; a = \sqrt{ x } + e^{\sqrt{b}}$	x, b
8	$y = \sin^3(a+b); a = t^3 + \sqrt{b}; b = \lg^2 x $	x, t
9	$y = \arctg^3 x^2; x = p+k; k = \sqrt{p+t^2}$	t, p
10	$y = \cos^2(a + \sin b); a = \sqrt{ x }; b = x^4 + m^2$	m, x
11	$y = \sin^3 a + \cos^2 x; a = c+k^2; c = \arctg x $	k, x
12	$y = e^{\sqrt{ x }} + \cos x; x = a+c^3; a = \sin^5 b$	b, c
13	$y = a \cos x - b \sin x; x = \sqrt[3]{a-b}; a = t^2 b$	t, b
14	$y = \sqrt{x} \sin a + \sqrt{b} \cos x; a = \lg x ; b = x + p^3$	x, p
15	$y = \lg a / \lg b; a = \sqrt{x^2 + b^2}; x = e^b + n$	n, b
16	$y = \ln x+t ; x = t^2 + p; t = \sqrt{m}$	m, p
17	$y = e^{a+b}; a = \lg t+b^2 ; t = b^2 + \sqrt{bx}$	b, x
18	$y = \sqrt[3]{x^2 + c^2}; x = e^{mk}; c = \cos^2 m + k^2$	k, m
19	$y = e^x + 5.8^c; c = a^2 + \sqrt{b}; a = b^3 + \ln b $	x, b
20	$y = x^3 / t^2; x = e^{\sqrt{p+a}}; t = p^3 + a^3$	a, p
21	$y = c^2 + \sqrt{ a }; c = \lg b ; a = (b+x)^3$	b, x
22	$y = \arctg^2 x ; x = t^3 + b^2; t = b^3 + e^{\sqrt{q}}$	q, b
23	$y = v^3 + \cos^2 w; v = \cos^2 a; w = \sqrt{a+ x }$	x, a
24	$y = x^2 + \sqrt[3]{ x }; x = \cos^2 b + \sin^2 a; a = \sqrt{b+t^2}$	b, t
25	$y = \sin^3 x + \cos x^2; x = \lg ct ; c = t^2 + \sqrt{a}$	t, a
26	$y = \lg^2 x+a ; x = \sqrt{a+b}; a = e^{t+b}$	t, b
27	$y = \arctg^3 p ; p = \sqrt{x^2 + a^2}; x = \sqrt{a} + \sqrt{b}$	a, b
28	$y = \sin^4(a^2 + b^2); a = \sqrt{b+t}; t = b^2 + k^3$	b, k
29	$y = \cos^3 x + a ; x = e^b; b = a + \sqrt{a+p^2}$	a, p
30	$y = \sin^4(a^2 + b^2); a = \sqrt{b+t}; t = b^2 + k^3$	b, k

Додаток А

Основні математичні функції

Функція C++	Опис	Клас Math
int abs (int i)	модуль (абсолютне значення) цілого числа $x - x $	Math:: Abs (x)
double fabs (double x)	модуль дійсного числа $x - x $	Math:: Abs (x)
double sqrt (double x)	корінь квадратний $-\sqrt{x}$	Math:: Sqrt (x)
double pow (double x, double y)	піднесення x до степеня $y - x^y$	Math:: Pow (x,y)
double exp (double x)	експонента e^x	Math:: Exp (x)
double log (double x)	натуральний логарифм $-\ln(x)$	Math:: Log (x)
double log10 (double x)	десятковий логарифм $-\lg(x)$	Math:: Log10 (x)
–	логарифм x за основою $N - \log_N(x)$	Math:: Log (x,N)
double cos (double x)	косинус $-\cos(x)$	Math:: Cos (x)
double sin (double x)	синус $-\sin(x)$	Math:: Sin (x)
double tan (double x)	тангенс $-\operatorname{tg}(x)$	Math:: Tan (x)
double acos (double x)	арккосинус $-\arccos(x)$	Math:: Acos (x)
double asin (double x)	арксинус $-\arcsin(x)$	Math:: Asin (x)
double atan (double x)	арктангенс $-\operatorname{arctg}(x)$	Math:: Atan (x)
double cosh (double x)	гіперболічний косинус $-(e^x + e^{-x})/2$	Math:: Cosh (x)
double sinh (double x)	гіперболічний синус $-(e^x - e^{-x})/2$	Math:: Sinh (x)
double ceil (double x)	округлення доверху: найменше ціле, не менше за x	Math:: Ceiling (x)
double floor (double x)	округлення донизу: найбільше ціле, не більше за x	Math:: Floor (x)

Додаток Б

Основні типи даних C++

Тип	Назва	Розмір, байт	Діапазон	Приклади можливих значень	Типи чисел
<code>char</code>	символьний (знаковий)	1	–128...127	'a', '\n', '9'	цілі
<code>unsigned char</code>	беззнаковий символьний	1	0...255	1, 233	
<code>short</code>	короткий цілий	2	–32 768...32 767	1, 153, –349	
<code>unsigned short</code>	беззнаковий короткий	2	0...65 535	0, 4, 65 000	
<code>int</code>	цілий (знаковий)	4*	–2 147 483 648... ...2 147 483 647	–30 000, 0, 690	
<code>unsigned int</code>	беззнаковий цілий	4	0...4 294 967 295	2 348, 60 864	
<code>long</code>	цілий (знаковий)	4	–2 147 483 648... ...2 147 483 647	–30 000, 0, 690	дійсні
<code>float</code>	дійсний одинарної точності	4	$3.4 \cdot 10^{-38}$... $3.4 \cdot 10^{38}$	3.23, –0.2 100.23, 12, –0.947, 0.0001, 6.34e–3, 4e5	
<code>double</code>	дійсний подвійної точності	8	$1.7 \cdot 10^{-308}$ $1.7 \cdot 10^{308}$		
<code>long double</code>	довгий дійсний	10	$3.4 \cdot 10^{-4932}$ $1.1 \cdot 10^{4932}$		
<code>bool</code>	логічний	1	false чи true	false(0), true(>=1)	
<code>enum</code>	перераховний	2 або 4			
<code>void</code>	порожній, без значення				
* – залежно від налагоджень компілятора й апаратних характеристик тип <code>int</code> може мати 4 або 2 байти.					

Список рекомендованої літератури

1. С++. Теорія та практика: навч. посіб. з грифом МОНУ/ [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. – Одеса : ВЦ ОНАЗ, 2011. – 587 с.
2. С++. Основи програмування. Теорія та практика: підручник / [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. – Одеса : Фенікс, 2010. – 544 с.
3. Страуструп Б. Язык программирования С++. Специальное издание ; пер. с англ. / Страуструп Б. – М. : ООО "Бином-Пресс", 2006. – 1104 с.
4. Стивен Прата. Язык программирования С++. Лекции и упражнения : учебник ; пер с англ. / Стивен Прата. – СПб.: ООО "ДиаСофтЮП", 2005. – 1104 с.
5. Зиборов В. В. MS Visual C++ 2010 в среде .NET. Библиотека программиста / Зиборов В. В. – СПб. : Питер, 2012. – 320 с.
6. Хортон А. Visual C++ 2010: полный курс.; пер. с англ. / Хортон А. – М. : ООО "И.Д. Вильямс", 2011. – 1216 с.
7. Довбуш Г. Ф. Visual C++ на примерах / Г.Ф. Довбуш, А.Д. Хомоненко ; под ред. проф. А.Д. Хомоненко. – СПб. : БХВ-Петербург, 2007. – 528 с.
8. Visual C++ .NET: пособие для разработчиков С++ / [А. Корера, С. Фрейзер, С. Джентайл и др.] – М. : ЛОРИ, 2003. – 398 с.
9. Дейтел Х. М. Как программировать на С++; пер с англ. / Х. М. Дейтел, П. Дж. Дейтел. – М. : ООО "Бином-Пресс", 2008. – 1456 с.
10. Трофименко О. Г. Алгоритмізація обчислювальних процесів і особливості програмування в С++ : метод. посібник. – Модуль 4 / Трофименко О. Г., Буката Л. М., Леонов Ю. Г. – Ч. 2. – Одеса : ІЦ ОНАЗ, 2009. – 93 с.

ЗМІСТ

Передмова	3
Структура дисципліни	4
Лабораторна робота № 1	
Знайомство з Visual C++. Способи введення-виведення даних	6
Теоретичні відомості	6
Питання та завдання для самоконтролю	13
Лабораторне завдання	13
Лабораторна робота № 2	
Програмування лінійних алгоритмів	14
Теоретичні відомості	14
Приклади програм з лінійною структурою в C++	20
Питання та завдання для самоконтролю	23
Лабораторне завдання	24
Лабораторна робота № 3	
Програмування розгалужень. Умовний оператор if	32
Теоретичні відомості	32
Приклади програм з розгалуженою структурою в C++	33
Питання для самоконтролю	36
Лабораторне завдання	36
Лабораторна робота № 4	
Програмування розгалужень. Оператор вибору варіантів switch	42
Теоретичні відомості	42
Приклади програм з розгалуженою структурою, з організованих за допомогою оператора switch	45
Питання для самоконтролю	49
Лабораторне завдання	49
Лабораторна робота № 5	
Програмування циклів. Оператор циклу з параметром for	59
Теоретичні відомості	59
Приклади програм з циклічною структурою, з організованих за допомогою оператора for	61
Питання для самоконтролю	63
Лабораторне завдання	63
Лабораторна робота № 6	
Циклічне опрацювання послідовностей чисел	67
Теоретичні відомості	67
Приклади програм	67
Лабораторне завдання	71

Лабораторна робота № 7

Вкладені цикли	74
Теоретичні відомості	74
Приклади проектів програм із вкладеними циклами	74
Питання та завдання для самоконтролю	76
Лабораторне завдання.....	76

Лабораторна робота № 8

Оператори циклу while та do-while	79
Теоретичні відомості	79
Приклади програм.....	80
Питання для самоконтролю	86
Лабораторне завдання.....	87

Лабораторна робота № 9

Організація функцій в C++	90
Теоретичні відомості	90
Приклади програм з функціями	92
Питання та завдання для самоконтролю	99
Лабораторне завдання.....	99

Додаток А

Основні математичні функції	103
--	-----

Додаток Б

Основні типи даних C++	104
-------------------------------------	-----

Список рекомендованої літератури	105
---	-----