

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 форматowanego 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;
```

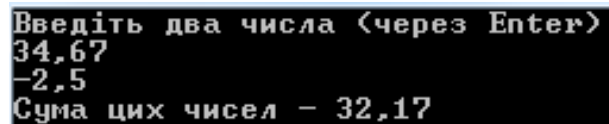


```
int main(array<System::String ^> ^args)
{
    Console::WriteLine(L"Здравствуй, мир!");
    return 0;
}
```

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

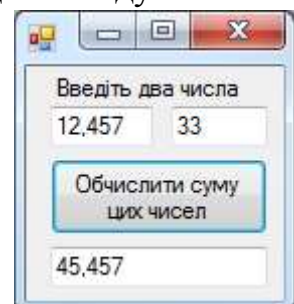
Для створення програмного проекту з формою слід запустити Visual Studio та виконати команду *Создать проект / Visual C++ / CLR / Приложение Windows Forms*. Не закриваючи вікно *Создать проект*, натиснути кнопку *Обзор*, вибрати теку для зберігання та задати ім'я програмного проекту. Натиснути кнопку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

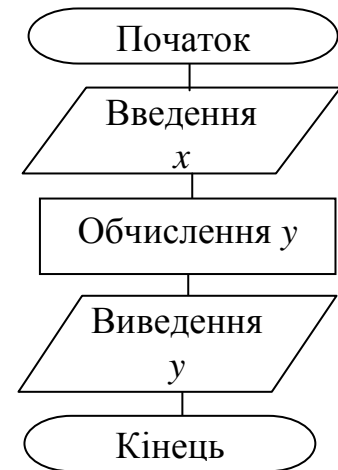
```
#include <iostream>
#include <math.h>
using namespace std;
```



```
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
```



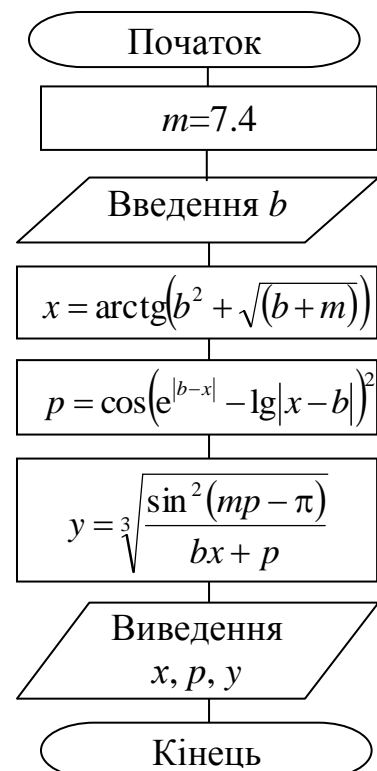
Приклад 2. Розробити схему алгоритму і програмний проект для обчислення $y = \sqrt[3]{\frac{\sin^2(mp - \pi)}{bx + p}}$; $x = \arctg(b^2 + \sqrt{(b+m)})$ і $p = \cos(e^{|b-x|} - \lg|x-b|)^2$. Значення $m = 7.4$ задати як константу, а довільне значення змінної b слід ввести.

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(x + \ln \sqrt{x^3 + 1.8})^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	$V23 = \cos^2\left(x + \sin(\sqrt{x^3 + \sqrt{x+1.5}} - \ln^2 x) + e^x\right)$

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)