Міністерство освіти України

Національний технічний університет "ХПІ"

Кафедра "Системний аналіз та інформаційно-аналітичні технології"

**Звіт**

**Лабораторна робота 4**

з дисципліни "Об’єктно орієнтоване програмування"

Виконав: студент групи КН-1223Б

Фільчак М.В.

Перевірив:

Івашко А.В.

Харків 2023

Зміст

[Мета роботи 3](#_Toc834616215)

[Завдання 3](#_Toc1718461542)

[Алгоритм завдання 3](#_Toc1695968789)

[Базові функції 3](#_Toc732785471)

[Розробка програми 4](#_Toc958162274)

[Результат роботи 5](#_Toc162074814)

[Висновок 5](#_Toc1887153977)

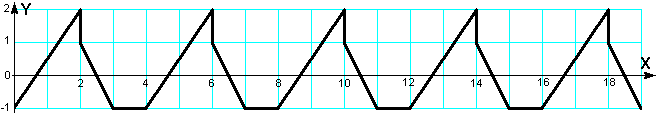
**Лабораторна робота №4**  
**Базові типи даних та введення — виводу**

Варіант 16

## **Мета роботи**

Метою лабораторної роботи є отримання практичних навичок в роботі з операторами циклів for(), while(), if() мови C++.

## **Завдання**



Для функції y = f(x), графік якої наведений у виданому індивідуальному завданні, вивести на екран значення y для x = 0, 0.25, 0.5 ... 19.75. Також необхідно вивести у консолі графік функції за допомогою засобів текстового режиму.

## **Алгоритм завдання**

З приведеного в індивідуальному завданні графіка функції видно, що функція є періодичною, значення періоду – 5. Нехай у заданий інтервал вкладаються 5 періодів функції. Отже, програма може містити в собі цикл, в якому 5 разів виконуватимуться одні й ті ж обчислення. У кожній ітерації цього циклу повинні перебиратися значення x від 0 до 3.75 з кроком 0.25, тобто, має бути вкладений цикл. У кожній ітерації вкладеного циклу обчислюється значення y для поточного значення x і виводяться результати.

Аналіз наведеного графіка показує, що кожен період його складається з трьох частин: на відрізку 0 <= x <= 2 маємо графік прямої заданий рівнянням y=1.5x-1, що потім на координаті x = 2 прямує вертикально вниз, на відрізку 2 <= x <= 3 – інша пряма задана рівнянням y=-2x+5, а на відрізку 3 <= x <= 4 – графік горизонтальної прямої. Отже, у вкладеному циклі мають виконуватися обчислення для всім частин графіку, що задані різними функціями.

## **Базові функції**

Оператор if() дозволяє проаналізувати вказану умову і здійснити виконання деякого фрагмента програми тільки при істинному результаті аналізу. Оператор if має дві форми:

if (вираз) оператор1;

if (вираз) оператор1; else оператор2;

Якщо в результаті обчислення виразу набуте значення true, то в обох формах оператора if виконується оператор1. Якщо в результаті обчислення виразу набуте значення false, тоді виконання оператора if, представленого в першій формі, закінчується, а в операторі, що має другу форму, виконується оператор2.

Оператор switch() дозволяє залежно від значення деякого виразу вибрати один з багатьох варіантів продовження програми. Оператор має наступний формат:

switch(вираз){

case(константний вираз1): оператор1; break;

case(константний вираз2): оператор2; break;

case(константний виразN): операторN; break;

default: операторN+1; }

Виконання оператора switch() починається з обчислення виразу, записаному в дужках. Цей результат послідовно порівнюється з результатами обчислення константних виразів при операторах case, і, якщо буде виявлена рівність результатів, то виконується оператор відповідного case. Якщо збіг результатів не виявлений, виконується оператор default. Якщо оператор default відсутній, то оператор switch пропускається.

## **Розробка програми**

Для початку необхідно підключити необхідні бібліотеки.

Ця бібліотека містить функції для операцій вводу-виводу.

#include <iostream>

Ця бібліотека декларує всі функції Windows API.

#include <Windows.h>

Далі у функцію main() додаємо підтримку кирилиці.

setlocale(LC\_ALL, "");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Через те, що у цій лабораторній роботі присутні додаткові завдання варто використовувати оператор switch() для створення початкового меню. Його можна зробити за допомогою змінної типу int, яку оператор switch() буде перевіряти на відповідність умовам операторів case: .

Далі програма починає виводити графік функцій. Після 5 повторень програма зупиняє цикл, а оператор у консолі може побачити графік, що виведений спеціальними символами ‘\*’.

Далі наведено частинний код програми:

double n, p;

double x, y;

printf\_s("|x | y|\n");

printf\_s("|-------|------------|\n");

for (n = 0; n < 5; n++) {

for (x = 0; x < 4; x += 0.25) {

if (x < 2) {

y = 1.5 \* x - 1;

}

else if (x < 3) {

y = -2 \* x + 5;

}

else {

y = -1;

}

printf\_s("| %5.2lf | %10.7lf |", x + n \* 4, y);

p = (y + 1) \* 10;

if (y - 1 - p \* 10 > 0.5) p++;

if (x == 2.0) {

printf\_s(" \* \*\n");

}

else {

for (; p > 0; p--) printf\_s(" ");

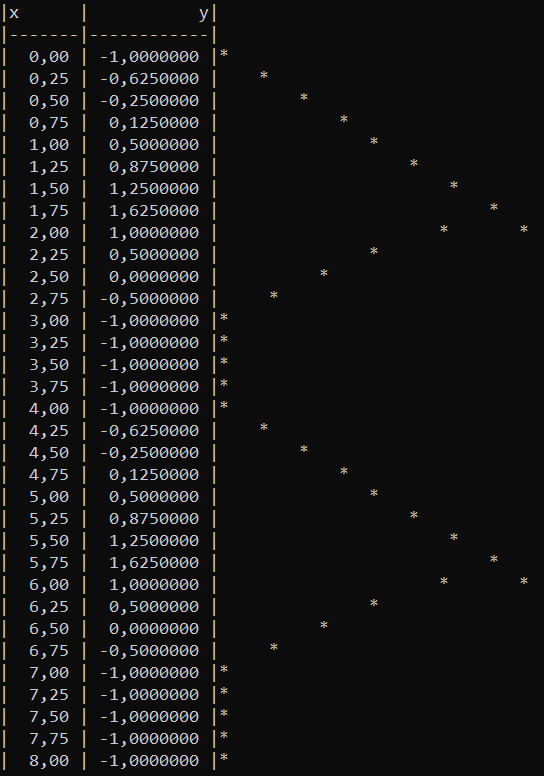
printf\_s("\*\n");

}

}

}

Результат роботи надано скріншотом:



## **Висновок**

Під час виконання цієї лабораторної роботи ми навчилися створювати контекстне меню для програм з декількох компонентів за допомогою оператора switch(), що перевіряє умову на відповідність операторам case: . Ці знання будуть корисними під час виконання лабораторних робіт з декількома завданнями.

Важливим також є отримання навичок роботи з умовними операторами if() та else, адже їх функціонал є досить потужним, вони будуть необхідними для виконання завдань. Для їх правильної роботи слід уважно перевірити коректність умови, а саме використання відповідних умовних позначень.

Іншим компонентом лабораторної роботи, що варто розглянути є зміна кольорів консолі та тексту в ній. Це можна робити різними способами, але найбільш компактним та зручним мені здалося використання функції #define для визначення кольорів, з допомогою цього можна спростити синтаксис у програмі. Це виглядає наступним чином:

#define RESET "\033[0m"

#define GREEN "\033[32m"

cout << GREEN << x << RESET << ";" << GREEN << y << endl;

Під час виконання одного з додаткових завдань цієї лабораторної роботи нам було необхідно конвертувати строку, що її ввів оператор на ціле число, для цього я використав функцію stoi(), порахувати кількість і суму цифр. Важливим є те, що під час введення цілого числа, що починається з 0, компілятор одразу відкидає його. Тому для виконання цього завдання необхідно одразу зчитувати строку, а не ціле число.