Міністерство освіти України

Національний технічний університет "ХПІ"

кафедра "Стратегічного управління і системного аналізу”

**Звіт**

**Лабораторна робота 4**

з дисципліни "Основи Програмування"

Виконав: студент групи КН-1223А

.Джим І.Р

Перевірив:

Івашко А.В.

Харків 2023

Зміст

[Зміст 2](#_Toc149300089)

[**Завдання** 3](#_Toc149300090)

[**1. Мета роботи** 3](#_Toc149300091)

[**2. Завдання для виконання** 3](#_Toc149300092)

[**Рішення** 4](#_Toc149300093)

[**Завдання 1** 4](#_Toc149300094)

[**Завдання 2** 4](#_Toc149300095)

[**Завдання 3** 5](#_Toc149300096)

[**Завдання 4** 7](#_Toc149300097)

[**Код проекту** 7](#_Toc149300098)

[**Висновок** 12](#_Toc149300099)

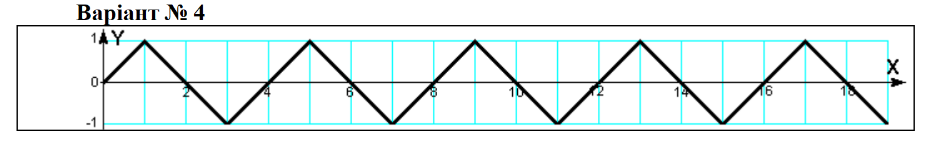
# **Завдання**

### **1. Мета роботи**

Метою лабораторної роботи є отримання практичних навичок в роботі з умовним оператором і розгалуженими алгоритмами в мові ***C***.

### **2. Завдання для виконання**

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. Необов’язкова для виконання додаткова вимога – отримати на екрані графік функції з використанням засобів текстового режиму.



3

# **Рішення**

Цей код містить програму з чотирьома завданнями, які можна вибрати за допомогою меню. Давайте розглянемо кожну функцію і її роботу детальніше.

# **Завдання 1**

1. `#define \_CRT\_SECURE\_NO\_WARNINGS`: Це директива препроцесора, яка вимикає попередження про потенційно небезпечні функції, які використовуються з бібліотеками стандартного C. В даному випадку, ця директива дозволяє використовувати деякі функції без отримання попереджень.

2. Підключення необхідних бібліотек: `iostream`, `stdio.h`, `Windows.h`, `math.h` для роботи зі стандартним вводом/виводом, функціями Windows API та математичними обчисленнями.

3. Визначення перерахування `MenuOptions`, яке представляє різні варіанти меню.

4. Функція `PrintMenuAndAskChoice(const HANDLE& h, MenuOptions& choice)`: Ця функція виводить меню на консоль з використанням дескриптора консолі `HANDLE` і запитує користувача обрати опцію зі списку `MenuOptions`.

5. Функція `PrintChartTask()`: Ця функція виводить графік на консоль. Графік складається з декількох рядків, де виводяться значення `x` та відповідні значення `y`, а також графічне представлення значення `y` за допомогою зірочок. Графік відображається у циклі з п'яти ітерацій.

# **Завдання 2**

1. `srand(time(0));`: Ця стрічка ініціалізує генератор випадкових чисел (`rand`) зі значенням, яке залежить від поточного часу (за допомогою функції `time(0)`). Це робить генерацію випадкових чисел більш непередбачуваною при кожному запуску програми, оскільки `rand()` в іншому виклику може відобразити інше значення.

4

2. `HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);`: Ця стрічка отримує дескриптор для стандартного виводу (консолі) і зберігає його в змінній `h`. Він потрібний для встановлення кольорів тексту в консолі.

3. `int from = 2, to = 9;`: Задає межі для множників, які будуть випадково генеруватися в межах цих чисел.

4. Починається цикл з 10 ітерацій, який генерує рівняння для вивчення таблиці множення:

a. `int firstMultiplier` і `int secondMultiplier` генерують випадкові числа множників у заданих межах (`from` і `to`).

b. `int answer` обчислює правильну відповідь, множення `firstMultiplier` на `secondMultiplier`.

c. Користувач вводить свою відповідь у `int userInput`. Поки вона не вірна, цикл продовжується.

d. Якщо відповідь невірна, виводиться помилка з червоним текстом, і оцінка зменшується на 1.

e. Якщо відповідь вірна, виводиться повідомлення про правильну відповідь з зеленим текстом, і оцінка збільшується на 1.

5. Після циклу виводиться оцінка, яку отримав користувач, і програма очищає консоль (`system("cls")`) для наступного запуску.

Цей код створює інтерактивну гру для вивчення таблиці множення, де користувач отримує оцінку на основі правильних відповідей. Функція `srand(time(0))` гарантує, що генератор випадкових чисел буде використовувати різні початкові значення при кожному запуску програми.

# **Завдання 3**

1. `srand(time(0));` - цей рядок ініціалізує генератор випадкових чисел `rand()` на основі поточного часу, щоб забезпечити випадковість чисел, які будуть

5

згенеровані пізніше.

2. `int from = 0, to = 100;` - визначає діапазон чисел, в якому комп'ютер буде загадувати число. У даному випадку, числа будуть від 0 до 100.

3. `int computerNumber = (rand() % (to - from + 1)) + from;` - комп'ютер вибирає випадкове число в заданому діапазоні та зберігає його у змінній `computerNumber`.

4. `int userNumber = -1;` - ініціалізуємо змінну `userNumber` значенням -1, яке буде використовуватися для введення числа користувачем.

5. `int tryCount = 0;` - змінна `tryCount` використовується для підрахунку кількості спроб, які користувач зробив, щоб вгадати число.

6. `HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);` та `SetConsoleTextAttribute(h, 15);` - ці рядки встановлюють колір тексту в консолі на білий (15).

7. Починається цикл `while`, який продовжується, доки `userNumber` не дорівнює `computerNumber`. У циклі користувач вводить свої варіанти чисел, і програма надає вказівки, чи число комп'ютера більше чи менше, ніж число користувача.

8. Після того, як користувач вгадав число, програма виводить повідомлення про кількість спроб, які були витрачені для вгадування.

9. `system("pause");` - зупиняє виконання програми, очікуючи введення користувачем.

6

10. `system("cls");` - очищає вміст консолі, щоб підготувати її до нової гри.

# **Завдання 4**

1. Отримує вказівник на стандартний вивід консолі (`HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);`) і встановлює колір тексту в білого (код 15).

2. Спочатку програма просить користувача ввести число з клавіатури (`cout << "Введіть число: ";`, `cin >> number;`).

3. Визначається кількість цифр у введеному числі і обчислюється сума цифр. Ця інформація буде використовуватися пізніше.

4. Шукається індекс останньої цифри у введеному числі (`endOfNumberIndex`). Це зроблено, перевіряючи символи в масиві `number`, поки не знайдеться символ '\0', який позначає кінець рядка.

5. Перевіряється, чи введене число є від'ємним (якщо перший символ - мінус). У разі негативного числа, виводиться повідомлення про невірне значення, і програма завершується.

6. Перевіряється, чи всі символи в рядку `number` є цифрами (від 0 до 9). Якщо знайдено символ, що не відповідає цим умовам, програма виводить повідомлення про помилку та завершується.

7. Після всіх перевірок виводяться цифри числа у зворотньому порядку разом із сумою цифр.

8. Нарешті, програма очищує консоль, очікує на натискання клавіші для закриття вікна консолі і завершує свою роботу.

Цей код призначений для введення та перевірки коректності введених чисел, виведення їх у зворотньому порядку та підрахунку суми цифр.

7

# **Код проекту**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <Windows.h>

#include <math.h>

using namespace std;

enum class MenuOptions { ClearConsole = -1, Exit = 0, PrintChartTask, GuesNumberTask, CheckMultiplyingTableTask, DigitSumTask };

void PrintMenuAndAskChoice(const HANDLE& h, MenuOptions& choice);

void PrintChartTask() {

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

short n;

double x, y;

short w;

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

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

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

for (x = 0; x <= 1; x += 0.25) {

y = x;

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

w = (y + 1) \* 10;

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

for (; w > 0; w--) printf(" ");

printf("\*\n");

}

for (x = 1; x <= 2; x += 0.25) {

y = 1 - (x - 1);

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

w = (y + 1) \* 10;

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

for (; w > 0; w--) printf(" ");

printf("\*\n");

}

}

system("pause");

system("cls");

}

void CheckMultiplyingTableTask() {

srand(time(0));

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

int from = 2, to = 9;

cout << "Вирішіть рівняння " << endl;

int mark = 0;

for (int i = 0; i < 10; i++) {

int firstMultiplier = (rand() % (to - from + 1)) + from;

int secondMultiplier = (rand() % (to - from + 1)) + from;

int answer = firstMultiplier \* secondMultiplier;

int userInput = -1;

while (userInput != answer) {

8

cout << firstMultiplier << " x " << secondMultiplier << " = ";

cin >> userInput;

if (userInput != answer) {

SetConsoleTextAttribute(h, 4);

cout << "Неправильна відповідь" << endl;

SetConsoleTextAttribute(h, 15);

mark--;

}

}

mark++;

SetConsoleTextAttribute(h, 10);

cout << "Правильна відповідь! " << endl;

SetConsoleTextAttribute(h, 15);

}

SetConsoleTextAttribute(h, 15);

cout << "Оцінка " << mark << "/10\n";

system("pause");

system("cls");

}

void GuesNumberTask()

{

srand(time(0));

int from = 0, to = 100;

int computerNumber = (rand() % (to - from + 1)) + from;

int userNumber = -1;

int tryCount = 0;

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(h, 15);

cout << "Від " << from << " до " << to << endl;

while (userNumber != computerNumber) {

cout << "Введіть число: ";

cin >> userNumber;

tryCount++;

if (userNumber < computerNumber) {

cout << "Більше " << endl;

}

else if (userNumber > computerNumber) {

cout << "Меньше " << endl;

}

}

cout << "Кількість спроб " << tryCount << " відповідь знайдена" << endl;

system("pause");

system("cls");

}

void DigitSumTask() {

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(h, 15);

char number[10];

cout << "Введіть число: ";

cin >> number;

int digitSum = 0;

int digitCount = 0;

int endOfNumberIndex = -1;

if (number[0] == '-') {

SetConsoleTextAttribute(h, 4);

cout << "Неправильне значення!" << endl;

system("pause");

system("cls");

return;

}

for (int i = sizeof(number) / sizeof(char) - 1; i > 0; i--) {

if (number[i] == '\0') {

endOfNumberIndex = i;

9

}

}

for (int i = 0; i <= endOfNumberIndex; i++) {

if (number[i] == '\0') {

continue;

}

int digit = int(number[i] - '0');

if (digit < 0 || digit>9) {

cout << "75 !" << endl;

return;

}

}

for (int i = endOfNumberIndex; i >= 0; i--) {

if (number[i] == '\0') {

continue;

}

int digit = int(number[i] - '0');

digitCount++;

digitSum += digit;

cout << digit;

}

cout << endl;

system("pause");

system("cls");

}

void PrintMenuAndAskChoice(const HANDLE& h, MenuOptions& choice)

{

SetConsoleTextAttribute(h, 15);

printf("Оберіть программу:");

SetConsoleTextAttribute(h, 6);

printf("\n%i", MenuOptions::PrintChartTask);

SetConsoleTextAttribute(h, 15);

printf(" - Графік");

SetConsoleTextAttribute(h, 6);

printf("\n%i", MenuOptions::GuesNumberTask);

SetConsoleTextAttribute(h, 15);

printf(" - Вгадай число");

SetConsoleTextAttribute(h, 6);

printf("\n%i", MenuOptions::CheckMultiplyingTableTask);

SetConsoleTextAttribute(h, 15);

printf(" - Таблиця множення");

SetConsoleTextAttribute(h, 6);

printf("\n%i", MenuOptions::DigitSumTask);

SetConsoleTextAttribute(h, 15);

printf(" - Обернуті числа");

SetConsoleTextAttribute(h, 6);

printf("\n%i", MenuOptions::Exit);

SetConsoleTextAttribute(h, 15);

printf(" - Вихід\nВарінт: ");

SetConsoleTextAttribute(h, 15);

scanf("%i", &choice);

}

int main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

MenuOptions choice;

while (true) {

10

PrintMenuAndAskChoice(h, choice);

switch (choice)

{

case MenuOptions::PrintChartTask:

PrintChartTask();

break;

case MenuOptions::GuesNumberTask:

GuesNumberTask();

break;

case MenuOptions::CheckMultiplyingTableTask:

CheckMultiplyingTableTask();

break;

case MenuOptions::DigitSumTask:

DigitSumTask();

break;

case MenuOptions::Exit:

return 0;

default:

SetConsoleTextAttribute(h, 4);

printf("Непраильна дія - %i\n", (int)choice);

SetConsoleTextAttribute(h, 7);

system("pause");

system("cls");

break;

}

}

}

11

# **Висновок**

В ході виконання лабораторної роботи у середовищі Visual Studio були отримані важливі навички програмування та роботи з інтегрованим середовищем розробки. За допомогою мови програмування, яка була використана в цій лабораторній роботі, вдалося створити програму, яка відповідає поставленим завданням.

Одним з ключових вмінь, які були розвинуті, є вміння розробляти і відлагоджувати код. Під час лабораторної роботи було необхідно виявити та виправити помилки в програмі, що сприяє покращенню розуміння процесу відлагодження.

Також, лабораторна робота дозволила ознайомитися з основними концепціями створення графічного інтерфейсу користувача, що є важливим аспектом розробки програм. Використання графічного інтерфейсу полегшило взаємодію з користувачем та зробило програму більш зручною та доступною.

Під час виконання лабораторної роботи було надано можливість вивчити та використовувати різноманітні бібліотеки та інструменти, які розширюють можливості Visual Studio. Це дозволяє покращити продуктивність та ефективність розробки.

Завдяки цій лабораторній роботі, ми також набули досвіду у спільній роботі над програмним проектом, що може бути корисним у майбутньому професійному житті.

У підсумку, виконання цієї лабораторної роботи дало нам можливість розширити свої знання та вміння в області програмування, відлагодження коду, створення графічного інтерфейсу та спільної роботи над проектами. Все це є важливими компонентами розвитку як професійного програміста.

12