

**Звіт**  
**з Лабораторної роботи №4**  
**за дисципліною «Програмування»**  
**студента групи ПА-24-1**  
**Сєроклин Святослава Павловича**  
**кафедра комп\*ютерних технологій, ДНУ,**  
**2023/2024**

## SUM

### 1. Постановка задачі

## SUM

### Завдання SUM

=====

Скласти програму SUM, яка:

- 1) вводить з клавіатури послідовність цілих чисел, що закінчується нулем;
- 2) знаходить суму всіх членів послідовності;
- 3) виводить на екран знайдене значення у вигляді:  
"SUM=..."

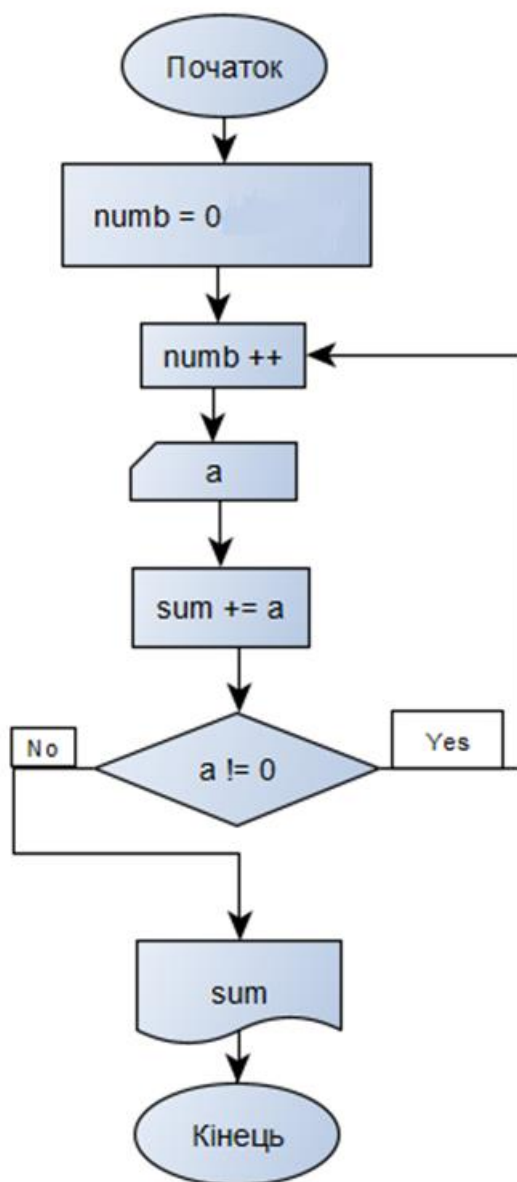
Приклад:

Введіть послідовність цілих чисел. Нуль – ознака кінця.

```
a[1]= 13
a[2]= 21
a[3]= 2
a[4]= 14
a[5]= -1
a[6]= -2
a[7]= 0
SUM=47
```

### 2. Опис розв'язку + блок-схема

У цій програмі ми маємо створити цикл, за допомогою якого ми будемо вводити елементи до тих пір, поки не введемо 0. Доки працює цикл ми будемо додавати до змінної sum кожне введене нами значення. Для цього зручно використати цикл do while.



### 3. Вихідний текст програми

```

#include <iostream>
#include <Windows.h> // Бібліотека мов
using namespace std;
int main()
{
    SetConsoleCP(1251); //для кирилиці

```

```
SetConsoleOutputCP(1251); //для кирилиці
```

```
double a, numb = 0, sum = 0;;
```

```
do
```

```
{
```

```
    numb++;
```

```
    cout << "a[" << numb << "] = ";
```

```
    cin >> a;
```

```
    sum += a;
```

```
} while (a != 0);
```

```
cout << "Сума елементів дорівнює: " << sum << endl;
```

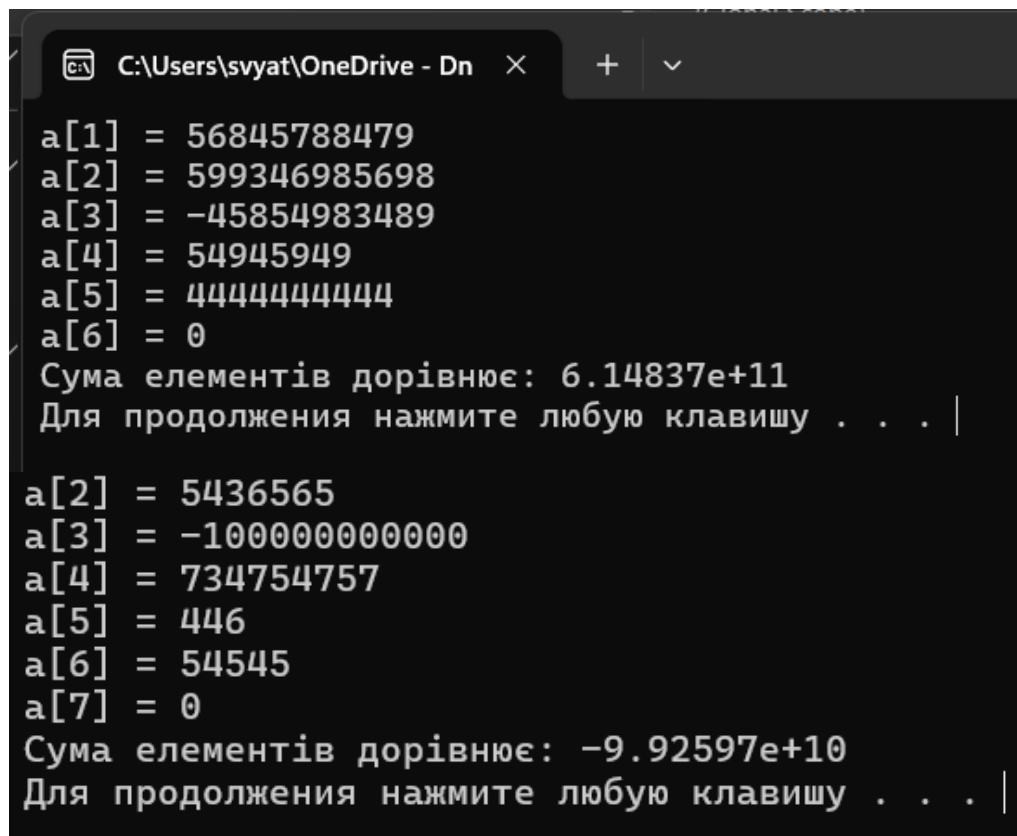
```
system("pause");
```

```
}
```

#### 4. Опис інтерфейсу програми

Запускаємо файл .exe . І вводимо елементи послідовності. Коли ми ввели всі потрібні елементи, вводимо 0.

#### 5. Опис тестових прикладів



```

C:\Users\svyat\OneDrive - Dn
a[1] = 56845788479
a[2] = 599346985698
a[3] = -45854983489
a[4] = 54945949
a[5] = 4444444444
a[6] = 0
Сума елементів дорівнює: 6.14837e+11
Для продовження натисніть будь-яку клавішу . . . |

a[2] = 5436565
a[3] = -1000000000000
a[4] = 734754757
a[5] = 446
a[6] = 54545
a[7] = 0
Сума елементів дорівнює: -9.92597e+10
Для продовження натисніть будь-яку клавішу . . . |
  
```

## PRODUCT

### 1. Постановка задачі

## PRODUCT

### Завдання PRODUCT

=====

Скласти програму PRODUCT, яка:

- 1) вводить з клавіатури послідовність цілих чисел, що закінчується нулем;
- 2) знаходить добуток усіх парних членів послідовності;
- 3) виводить на екран обчислений добуток у вигляді: "ДОБУТОК = ...".

Приклад:

Введіть послідовність цілих чисел. Нуль – ознака кінця.

a[1]= 13

a[2]= 21

a[3]= 2

a[4]= 14

a[5]= -1

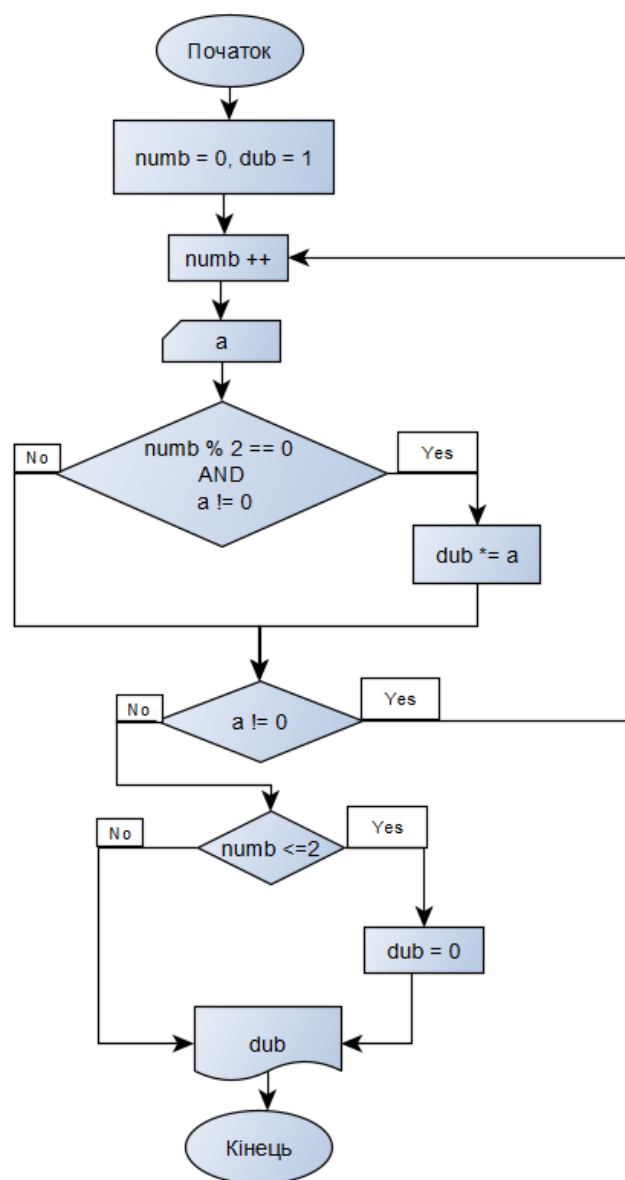
a[6]= -2

a[7]= 0

ДОБУТОК=-56

### 2. Опис розв'язку + блок-схема

Складемо програму, що буде використовувати цикл, який завершиться тільки тоді, коли введене значення змінної буде 0. Визначимо парність елементу за допомогою остачі від ділення його номеру на 2. Добуток підрахуємо за допомогою змінної, чиє початкове значення буде 1, і котра буде множити парні елементи послідовності на саму себе.



### 3. Вихідний текст програми

```

#include <iostream>
#include <Windows.h> // Бібліотека мов
using namespace std;
int main()
{
    SetConsoleCP(1251); //для кирилиці
    SetConsoleOutputCP(1251); //для кирилиці
    int a, numb = 0;

```

```

double dub = 1;
do
{
    numb++;
    cout << "a[" << numb << "] = ";
    cin >> a;
    if (numb % 2 == 0)
    {
        if (a != 0)
        {
            dub *= a;
        }
    }
}

} while (a != 0);
if (numb <= 2)
{
    dub = 0;
}
cout << "ДОБУТОК = " << dub << endl;
system("pause");
}

```

#### 4. Опис інтерфейсу програми

Запускаємо файл .exe . І вводимо елементи послідовності. Коли ми ввели всі нам потрібні елементи, то вводимо 0. Це запустить підрахунок добутку парних членів послідовності.

#### 6. Опис тестових прикладів

```

a[1] = 56
a[2] = 453
a[3] = 67
a[4] = 0
ДОБУТОК = 453
Для продолжения нажмите любую клавишу . . . |

```

```

a[1] = 37247634
a[2] = 323
a[3] = 3454
a[4] = 2
a[5] = 54
a[6] = 1
a[7] = 0
ДОБУТОК = 646
Для продолжения нажмите любую клавишу . . . |

```

```

a[1] = 54
a[2] = 432945
a[3] = 42934
a[4] = 43242324
a[5] = 42334
a[6] = 3423443
a[7] = 2432
a[8] = 243432324
a[9] = 0
ДОБУТОК = 1.56021e+28
Для продолжения нажмите любую клавишу . . . |

```

```

a[1] = 1
a[2] = 0
ДОБУТОК = 0
Для продолжения нажмите любую клавишу . . . |

```



## COUNTODD

### 1. Постановка задачі

#### COUNTODD

Задача "CountOdd"

=====

Скласти програму CountOdd, яка:

1. Вводить із клавіатури послідовність цілих чисел, що закінчується нулем.
2. Знаходить скільки серед них непарних.
3. Як тільки вводиться число 0, програма припиняє роботу та видає результат підрахунку на екран.

Приклад :

Введіть число = 11

Введіть число = 10

Введіть число = 3

Введіть число = 1

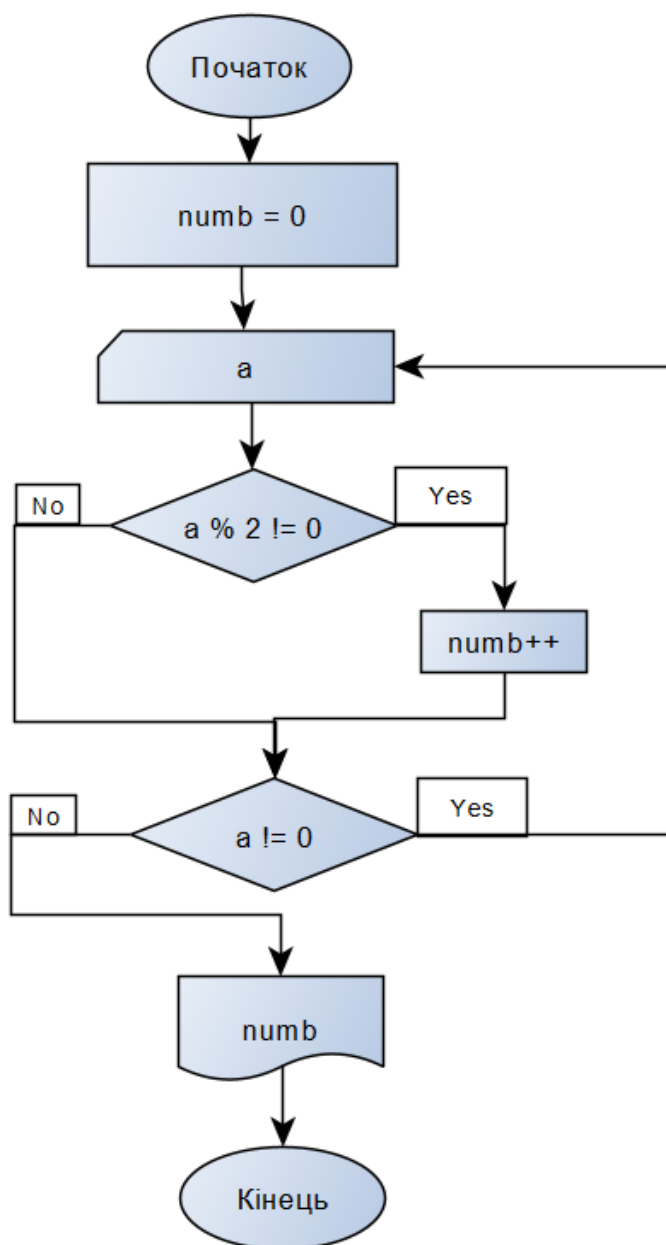
Введіть число = 2

Введіть число = 0

Зустрілося 3 непарних числа.

### 2. Опис розв'язку + блок-схема

Створимо цикл, що працюватиме, доки користувач не введе 0. Коли користувач буде вводити значення змінної, ми перевіримо її непарність за допомогою остачі від ділення на 2. Для підрахунку непарних значень створимо окрему змінну, що буде збільшувати своє значення на 1, коли число буде непарним.



### 3. Вихідний текст програми

```

#include <iostream>
#include <Windows.h> // Бібліотека мов
using namespace std;
int main()
{
    SetConsoleCP(1251); //для кирилиці
    SetConsoleOutputCP(1251); //для кирилиці
    int a, numb = 0;

```

```

do
{
    cout << "Введіть число = ";
    cin >> a;
    if (a % 2 != 0)
    {
        ++numb;
    }

} while (a != 0);

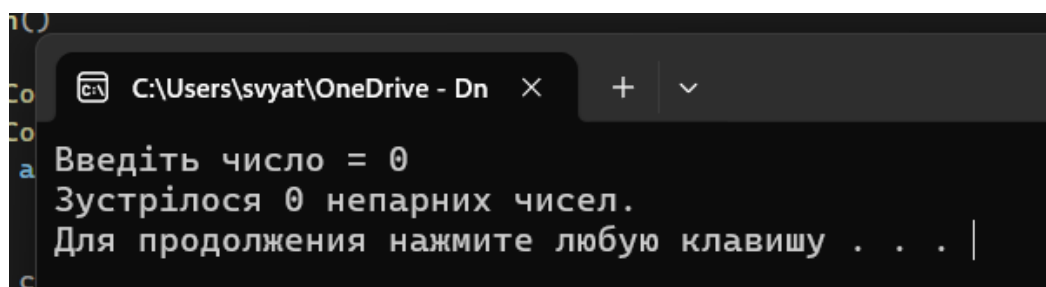
cout << "Зустрілося " << numb << " непарних чисел." << endl;
system("pause");
}

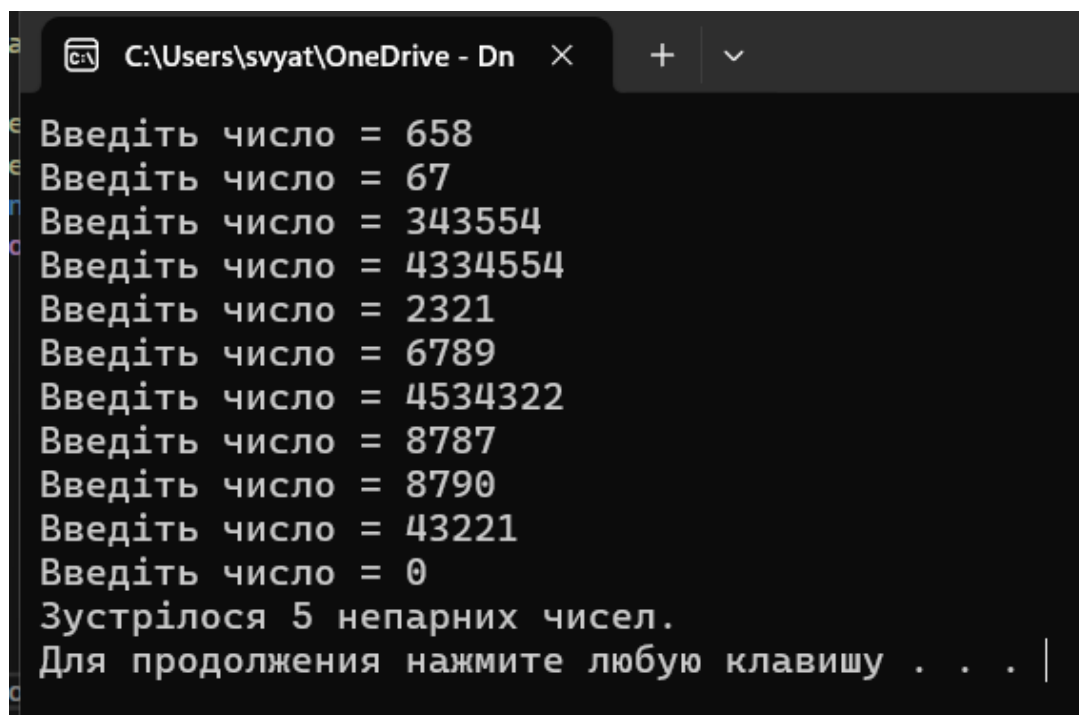
```

#### 4. Опис інтерфейсу програми

Запускаємо файл .exe . Користувач вводить всі потрібні йому значення, після чого в якості останньої змінної вводить 0.

#### 5. Опис тестових прикладів





```
C:\Users\svyat\OneDrive - Dn  X + v
Введіть число = 658
Введіть число = 67
Введіть число = 343554
Введіть число = 4334554
Введіть число = 2321
Введіть число = 6789
Введіть число = 4534322
Введіть число = 8787
Введіть число = 8790
Введіть число = 43221
Введіть число = 0
Зустрілося 5 непарних чисел.
Для продовження натисніть будь-яку клавішу . . . |
```

## Parrot 2

### 1. Постановка задачі

#### Parrot2

Завдання Parrot2

=====

Складіть програму PARROT2, яка:

1. Просить користувача ввести з клавіатури один рядок якогось тексту.

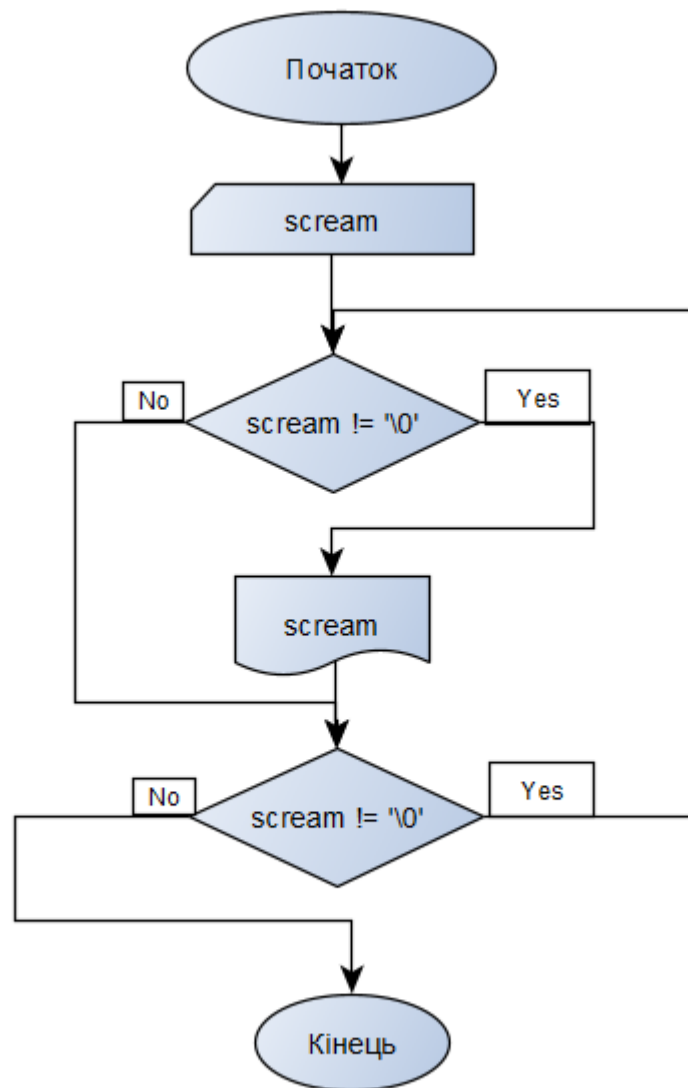
---

2. Виводить цей рядок на екран.

3. Повторює ці дії до того часу, поки користувач не введе порожній рядок (тобто не натисне просто ENTER).

### 2. Опис розв'язку + блок-схема

Створимо програму, в якій діятиме цикл. Доки ми не ввели пустий рядок (зробимо це в коді за допомогою `\0`), програма буде діяти. Ми зможемо вводити цілий рядок за допомогою `cin.getline`. Без різниці, якому типу буде належати введена змінна.



### 3. Вихідний текст програми

```

#include<iostream>
#include<Windows.h>
using namespace std;
int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char scream[51];
    cout << "Я - Папага Петті! Скажи мені щось, і я це повторю. (Але не
    більше 50 символів, друже)" << endl;
  
```

```
do
{
    cin.getline(scream, 51);
    if (scream[0] != '\0')
    {
        cout << scream << endl << endl;
    }
}
while (scream[0] != '\0');
system("pause");
}
```

#### 4. Опис інтерфейсу програми

Користувач вводить своє повідомлення, після чого програма його виводить на консоль. Коли користувач хоче завершити програму, то він просто нічого не вводить і натискає Enter.

## 5. Опис тестових прикладів

```

C:\Users\svyat\OneDrive - Dn
Я - Папага Петті! Скажи мені щось, і я це повторю. (Але не більше 50 символів, друже)
лрлрщпщимщимм
лрлрщпщимщимм

ппвпвлпвлп пвлпавпппала
ппвпвлпвлп пвлпавпппала

ЧАЙ ПОНАД УСЕ!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ЧАЙ ПОНАД УСЕ!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

0
0

Для продолжения нажмите любую клавишу . . .

```

## MENU 2

### 1. Постановка задачі

#### MENU2

Завдання MENU2

=====

Скласти програму MENU2, яка:

1) виводить на екран наступний текст:

"Працює інформаційна система ФПМ:

1. Новини дня

2. Анекдот тижня

3. Автора!

0. Вихід

Введіть відповідний номер та натисніть ENTER."

2) вводять із клавіатури номер;

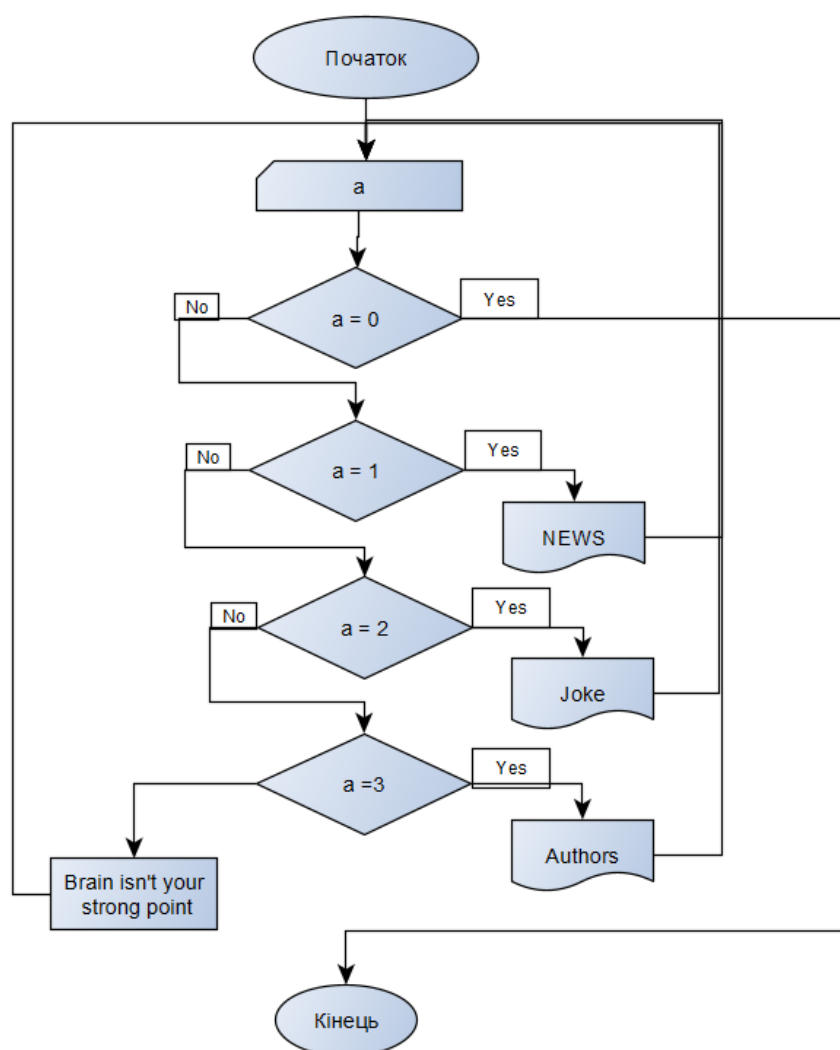
3) виводить на екран відповідну інформацію;

4) продовжує циклічно виконувати вищезгадані дії доки користувач не вибере "Вихід".

### 2. Опис розв'язку + блок-схема

Створимо цикл, в середині якого буде змінна *a*, котру буде вводити користувач доти, доки значення цієї змінної не стане 0.





### 3. Вихідний текст програми

```

#include <iostream>
using namespace std;
#include <Windows.h>
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int a;
    cout << "Працює інформаційна система факультету прикладної
математики:\n";
    cout << "0. Вихід\n";

```

```

cout << "1. Новина дня\n";
cout << "2. Анекдот тижня\n";
cout << "3. Автори !!!\n";
cout << "Введіть відповідний номер і натисніть ENTER \n";
    do
    {
        cin >> a;
        switch (a)
        {
            case 0:
                cout << "Ви завершили програму." << endl;
                break;
            case 1:
                cout << "Відомий український стрімер 'Папіч'
розпочав проходити Balatro." << endl;
                break;
            case 2:
                cout << "На даху зустрічаються два коти –
київський та одеський." << endl <<
                    "– Ну що, пом'яуємо? – пропонує
київський." << endl <<
                    "– Таки мяу." << endl;
                break;
            case 3:
                cout << "Володимир Кшиштовський, Леонід
Байгорода, Максим Свинера, Ічіго Куросакі" << endl;
                break;
            default:

```

```

        cout << "Ти зовсім тупенький(-ка)? Є всього 3
варіанти" << endl;
    }
} while (a != 0);
system("pause");
}

```

#### 4. Опис інтерфейсу програми

Користувач запускає файл .exe . Далі він вводить будь-яке ціле число. В залежності від числа він отримує результат.

#### 5. Опис тестових прикладів

```

C:\Users\svyat\OneDrive - Dn
Працює інформаційна система факультетуприкладної математики:
0. Вихід
1. Новина дня
2. Анекдот тижня
3. Автори !!!
Введіть відповідний номер і натисніть ENTER
56
Ти зовсім тупенький(-ка)? Є всього 3 варіанти
3
Володимир Кшиштовський, Леонід Байгорода, Максим Свинера, Ічіго Куросакі
-9
Ти зовсім тупенький(-ка)? Є всього 3 варіанти
54
Ти зовсім тупенький(-ка)? Є всього 3 варіанти
1
Відомий український стрімер 'Папіч' розпочав проходити Balatro.
2
На даху зустрічаються два коти – київський та одеський.
– Ну шо, пом'яукаємо? – пропонує київський.
– Таки мяу.
3
Володимир Кшиштовський, Леонід Байгорода, Максим Свинера, Ічіго Куросакі
0
Ви завершили програму.
Для продовження натисніть будь-яку клавішу . . . |

```

## MAX

### 1. Постановка задачі

## MAX

#### Завдання MAX

=====

Скласти програму MAX, яка:

- 1) вводить з клавіатури послідовність цілих чисел, яка завершується нулем;
- 2) знаходить найбільший із усіх членів послідовності;
- 3) виводить на екран знайдене значення у вигляді: "MAX = ...".

Приклад:

Введіть послідовність цілих чисел. Нуль – ознака кінця.

a[1]= 13

a[2]= 21

a[3]= 2

a[4]= 14

a[5]= -1

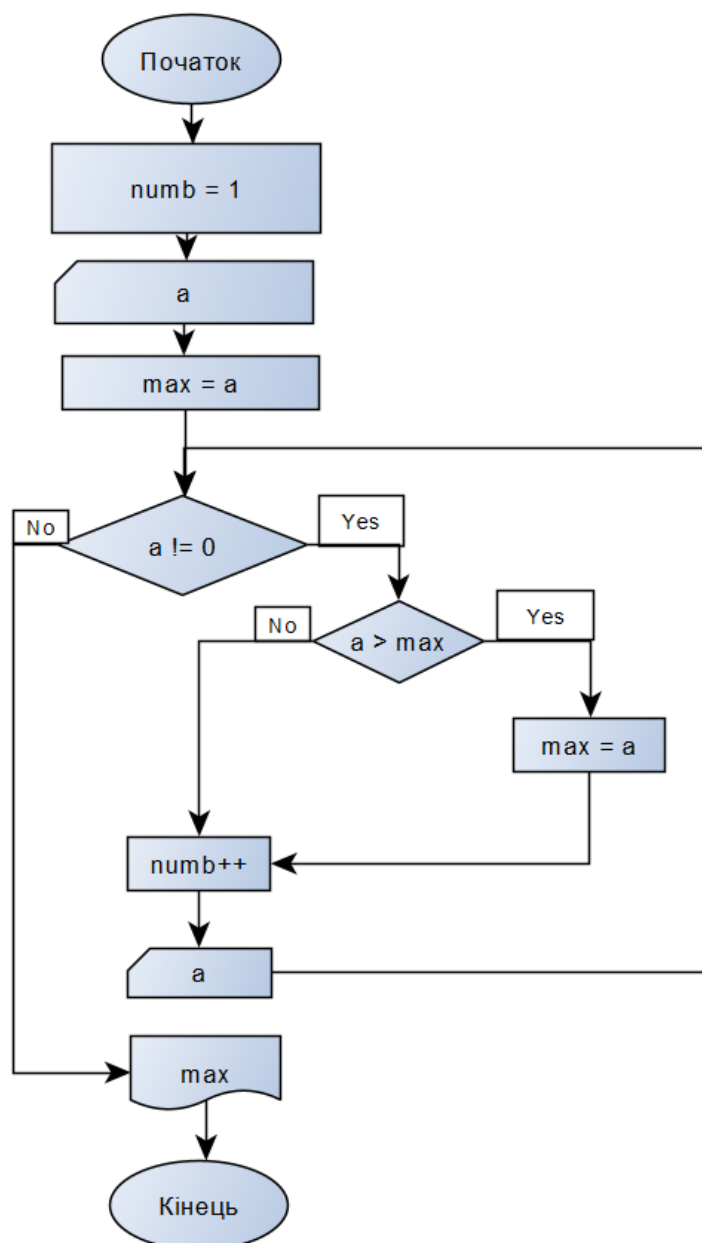
a[6]= -2

a[7]= 0

MAX=21

### 2. Опис розв'язку + блок-схема

Створимо програму, в котрій ми будемо вводити значення змінної, котре програма буде вважати максимальним. Зробимо цикл, в котрому ми будемо вводити значення змінної доти, поки не введемо 0. Програма буде порівнювати нове значення змінної з максимальним і, якщо значення змінної більше, то тепер саме воно буде максимальним.



### 3. Вихідний текст програми

```

#include <iostream>
#include <Windows.h> // Бібліотека мов
using namespace std;
int main()
{
    SetConsoleCP(1251); //для кирилиці
    SetConsoleOutputCP(1251); //для кирилиці
    int a, numb = 1;

```

```

cout << "a[" << numb << "]" = ";
cin >> a;
int max = a;
while (a != 0)
{
    if (a > max) {
        max = a;
    }
    numb++;
    cout << "a[" << numb << "]" = ";
    cin >> a;
}
cout << "MAX = " << max << endl;

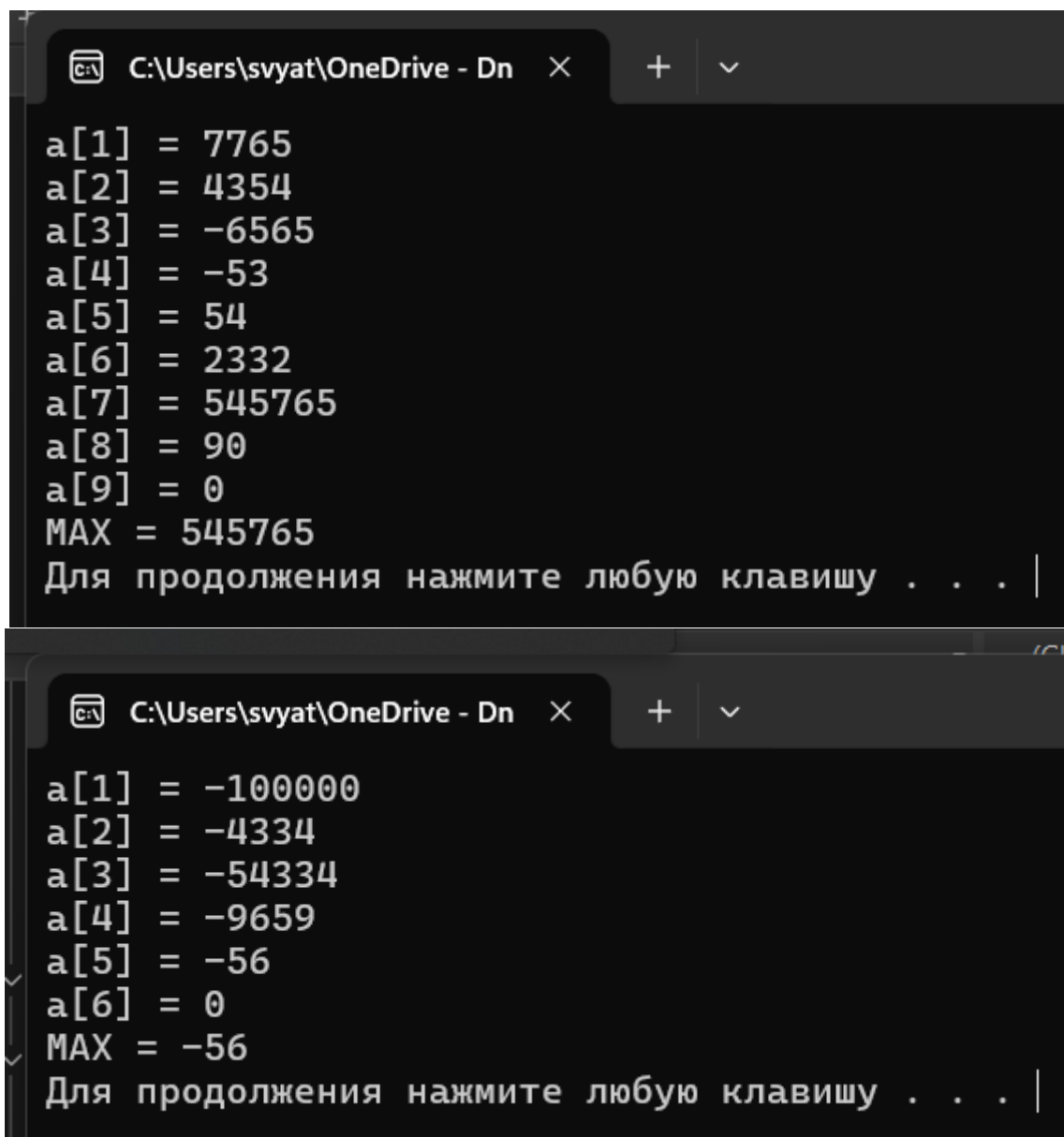
system("pause");
}

```

#### 4. Опис інтерфейсу програми

Запускаємо файл .exe .Користувач має буде вводити всі йому потрібні елементи послідовності. Коли він закінчить, то введе 0 останнім елементом.

## 5. Опис тестових прикладів



The image shows two screenshots of a C++ IDE window titled "C:\Users\svyat\OneDrive - Dn". The first screenshot displays the values of an array 'a' and its maximum value 'MAX'. The second screenshot displays the values of an array 'a' and its maximum value 'MAX'.

```
a[1] = 7765
a[2] = 4354
a[3] = -6565
a[4] = -53
a[5] = 54
a[6] = 2332
a[7] = 545765
a[8] = 90
a[9] = 0
MAX = 545765
Для продолжения нажмите любую клавишу . . . |
```

```
a[1] = -100000
a[2] = -4334
a[3] = -54334
a[4] = -9659
a[5] = -56
a[6] = 0
MAX = -56
Для продолжения нажмите любую клавишу . . . |
```

## Fibonacci

### 1. Постановка задачі

#### Fibonacci.txt

Завдання Fibonacci (FIBO)

=====

Послідовність чисел Фібоначчі - це числа  $f_0, f_1, f_2, f_3 \dots$  :

$f[0]=0$

$f[1]=1$

$f[N]=f[N-1]+f[N-2]$ , якщо  $N \geq 2$ .

Легко бачити, що з цього визначення виходить наступна послідовність чисел:

0 1 1 2 3 5 8 13 21 ...

Складіть програму FIBO, яка вводить з клавіатури число  $N$ , обчислює число Фібоначчі з номером  $N$  (тобто  $f[N]$ ) та виводить його на екран.

Приклад роботи програми:

$N = 7$

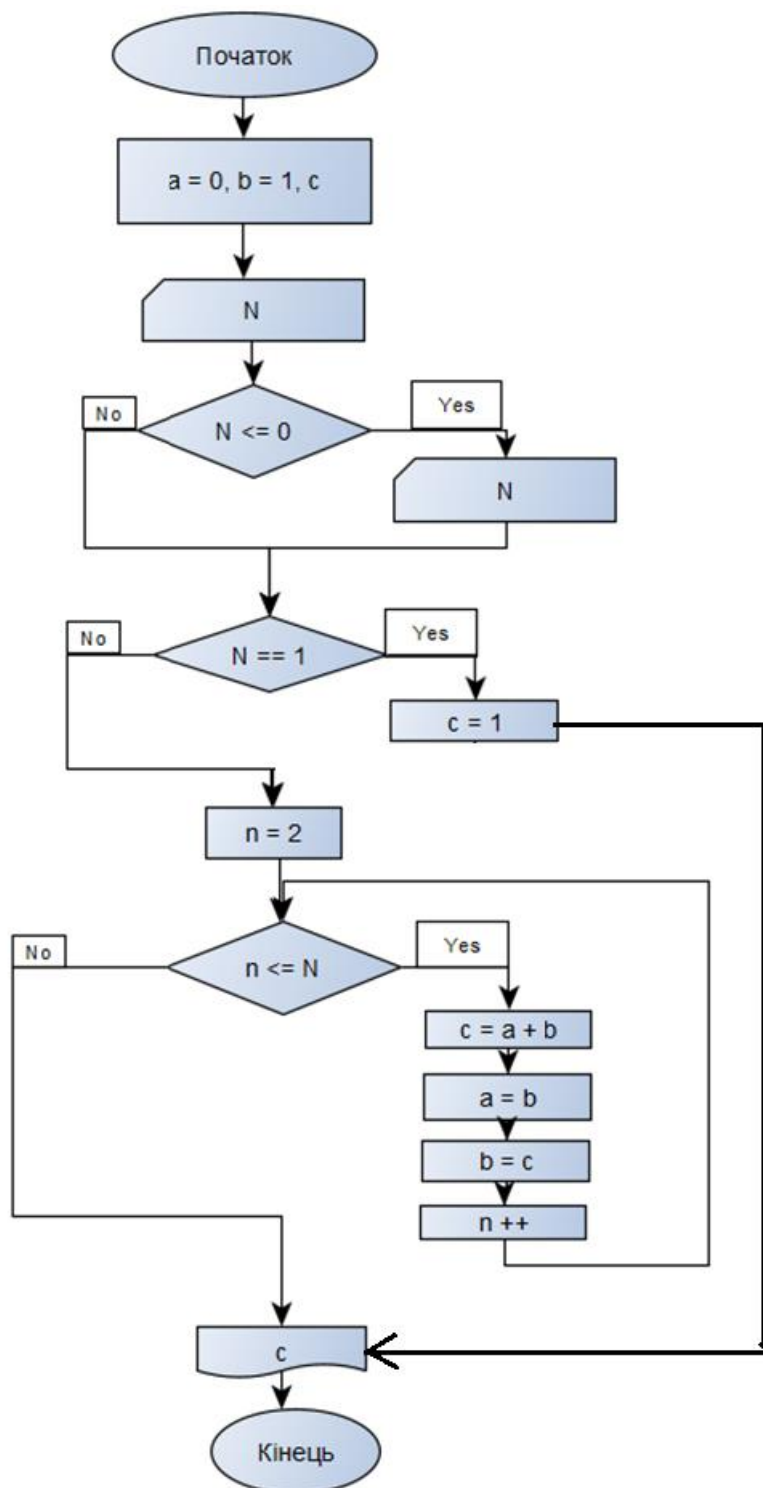
$F(7) = 13$

### 2. Опис розв'язку + блок-схема

Створимо цикл, що буде виконуватись певну кількість разів. Вона буде залежати від введеного нами номера елемента послідовності. Якщо



введений номер – це від’ємне число чи 0, то дамо можливість користувачеві знову ввести номер елементу.



### 3. Вихідний текст програми

```
#include <iostream>
```

```
#include <Windows.h> // Бібліотека мов
```

```

using namespace std;

int main()
{
    SetConsoleCP(1251); //для кирилиці
    SetConsoleOutputCP(1251); //для кирилиці
    long long int N, a = 0, b = 1, c;
    cout << "N = ";
    cin >> N;
    while( N <= 0)
    {
        cout << "N має бути натуральним числом." << endl << "N = ";
        cin >> N;
    }
    if(N == 1)
    {
        c = 1;
    }
    else
    {
        {
            for (int n = 2; n <= N; n++)
            {
                c = a + b;
                a = b;
                b = c;
            }
        }
    }
}

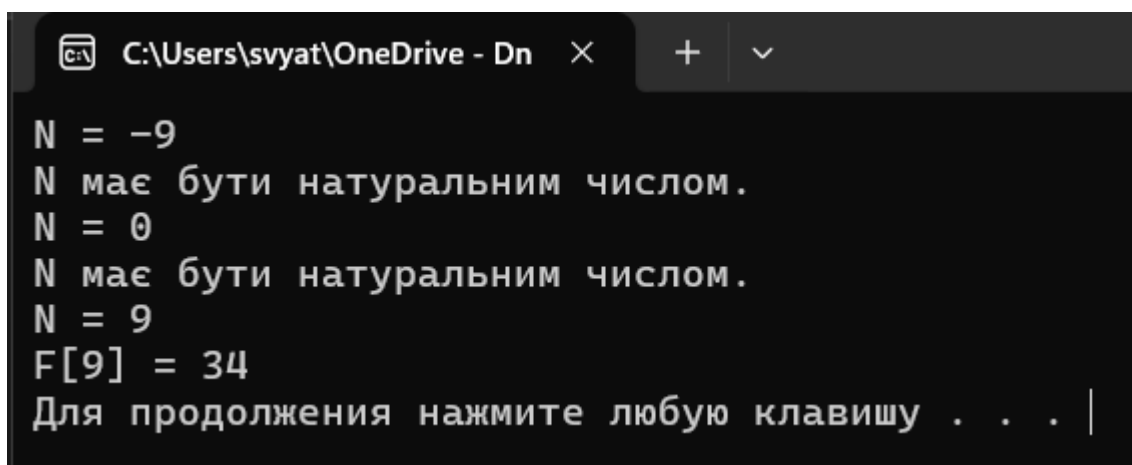
```

```
cout << "F[" << N << "] = " << c << endl;
system("pause");
}
```

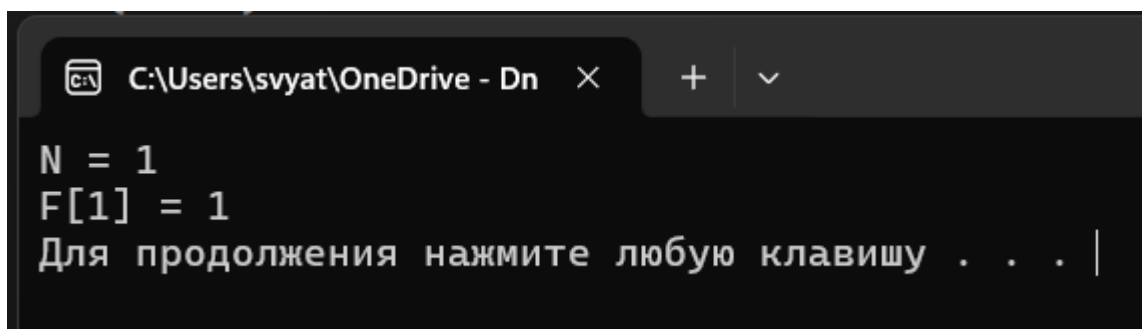
#### 4. Опис інтерфейсу програми

Відкриваємо файл .exe . Вводимо будь-яке значення N. Якщо введене значення менше або дорівнює 0, то вводимо N знову. На консоль виводиться елемент з відповідним номером

#### 5. Опис тестових прикладів



```
C:\Users\svyat\OneDrive - Dn
N = -9
N має бути натуральним числом.
N = 0
N має бути натуральним числом.
N = 9
F[9] = 34
Для продовження натисніть будь-яку клавішу . . . |
```



```
C:\Users\svyat\OneDrive - Dn
N = 1
F[1] = 1
Для продовження натисніть будь-яку клавішу . . . |
```

## НІТ 2

### 1. Постановка задачі

#### НІТ2

Завдання НІТ2

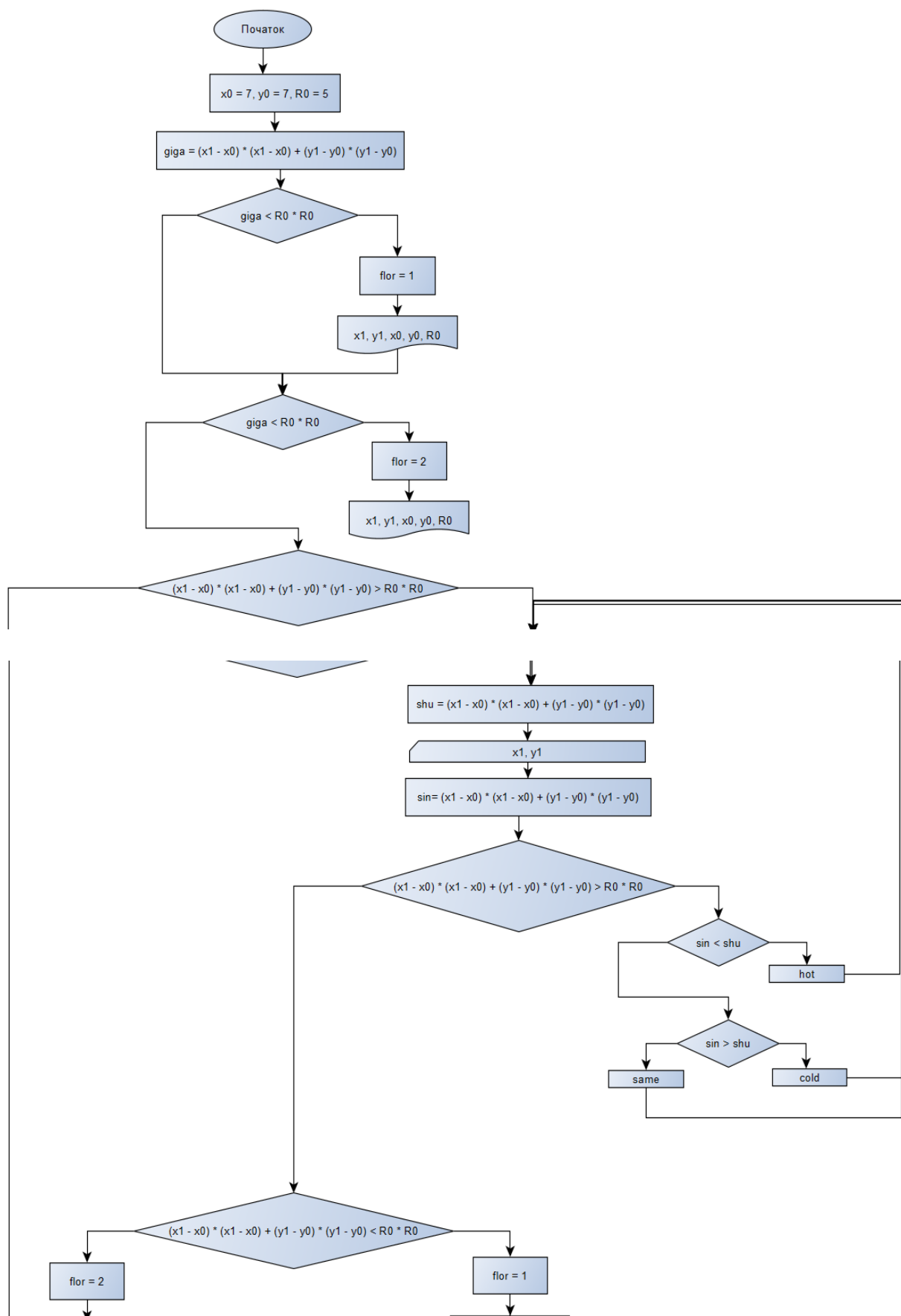
=====

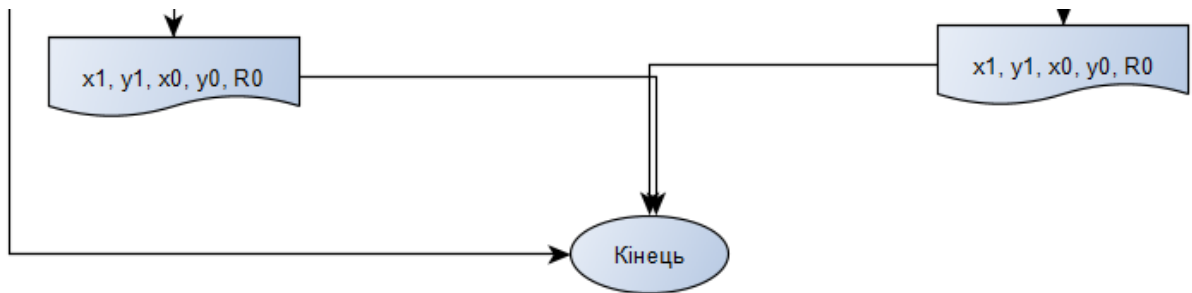
Скласти програму НІТ2, яка:

0. "Задумує" коло  $O$  з центром у точці  $(x_0, y_0)$  та радіусом  $R$ .
1. Вводить з клавіатури координати  $(x, y)$  точки  $A$  на площині.
2. Перевіряє, чи потрапляє точка  $A$  всередину (або на кордон) кола  $O$ .
3. Доки точка не потрапила всередину, виводить повідомлення "Тепліше" або "Холодніше" в залежності від того, ближче або далі від центру кола потрапила нова точка порівняно з попередньою, та повторює пункти 1, 2, 3.
4. Виводить результат на екран у вигляді:  
"Точка  $(x, y)$  потрапила в коло з центром у точці  $(x_0, y_0)$  радіуса  $R$ ",  
підставляючи результат замість  $R, x_0, y_0, x, y$  їх значення.

### 2. Опис розв'язку + блок-схема

Створимо програму, що буде вираховувати, чи знаходиться точка в колі за формулою. Якщо точка буде не в колі, то створимо цикл, де користувач буде вводити числа доти, доки точка за введеними координатами не опиниться в колі. У нас будуть 2 змінні, одна з яких прийме значення уже введених координат, а інша – координат, що ми введемо. Порівнюючи ці змінні, ми визначимо, чи ближче чи далі від центру кола ми знаходимось.





### 3. Вихідний текст програми

```

#include <iostream> // Бібліотека вводу і виводу
#include <Windows.h> // Бібліотека мов
using namespace std;
int main() // тіло програми
{
    SetConsoleCP(1251); //для кирилиці
    SetConsoleOutputCP(1251); //для кирилиці
    float giga, x0 = 7, y0 = 7, R0 = 5, x1, y1, flor;
    cout << "Введи координати точки А" << endl;
    cout << "x1 = "; cin >> x1;
    cout << "y1 = "; cin >> y1;
    giga = (x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0);
    if (giga < R0 * R0)
    {
        flor = 1;
        cout << "Точка А (" << x1 << ";" << y1 << ") потрапляє в коло з центром (" << x0 << ";" << y0 << ") і радіусом " << R0 << "." << endl;
    }
    else if (giga == R0 * R0)
    {
        flor = 2;
        cout << "Точка А (" << x1 << ";" << y1 << ") лежить на кордоні кола з центром (" << x0 << ";" << y0 << ") і радіусом " << R0 << "." << endl;
    }
}
  
```

```

}
else
{
    cout << "Точка А не потрапила. Спробуй ще раз." << endl;

    while ((x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0) > R0 * R0)
    {
        double shu, sin;
        shu = (x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0);
        cout << "x1 = "; cin >> x1;
        cout << "y1 = "; cin >> y1;
        sin = (x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0);
        if ((x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0) > R0 * R0)
        {
            if (sin < shu)
            {
                cout << "Тепліше" << endl;

            }
            else if (sin > shu)
            {
                cout << "Холодніше." << endl;
            }
            else if (sin == shu)
            {
                cout << "Відстань від центру колу не змінилась. " << endl;
            }
        }
    }
}

```

```

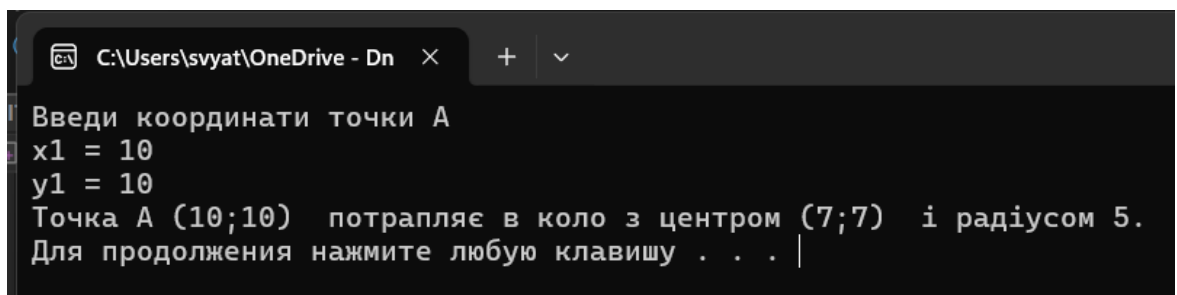
    }
    if ((x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0) < R0 * R0)
    {
        flor = 1;
        cout << "Точка А (" << x1 << ";" << y1 << ") потрапляє в коло з
центрoм (" << x0 << ";" << y0 << ") і радіусом " << R0 << "." << endl;
    }
    else if ((x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0) == R0 * R0)
    {
        flor = 2;
        cout << "Точка А (" << x1 << ";" << y1 << ") лежить на кордоні кола
з центрoм (" << x0 << ";" << y0 << ") і радіусом " << R0 << "." << endl;
    }
}
system("pause");
}

```

#### 4. Опис інтерфейсу програми

Користувач вводить 2 числа, що задають координати точки А. Якщо ця точка в коло (або на його кордон), то користувач побачить цю інформацію. Якщо користувач введе точку, що не належить колу, то він отримає можливість знову ввести точку. Якщо нова точка буде знаходитись ближче до центру кола, то він отримає повідомлення «Тепліше», а якщо далі, то «Холодніше».

#### 5. Опис тестових прикладів



```

C:\Users\svyat\OneDrive - Dn
Введи координати точки А
x1 = 10
y1 = 10
Точка А (10;10) потрапляє в коло з центрoм (7;7) і радіусом 5.
Для продовження натисніть будь-яку клавішу . . . |

```



```
C:\Users\svyat\OneDrive - Dn  ×  +  v

Введи координати точки A
x1 = 34
y1 = 34
Точка A не потрапила. Спробуй ще раз.
x1 = 60
y1 = 60
Холодніше.
x1 = 60
y1 = 60
Відстань від центру колу не змінилась.
x1 = 23
y1 = 23
Тепліше
x1 = -9
y1 = -9
Відстань від центру колу не змінилась.
x1 = -6
y1 = -6
Тепліше
x1 = 0
y1 = 0
Тепліше
x1 = 12
y1 = 7
Точка A (12;7) лежить на кордоні кола з центром (7;7) і радіусом 5.
Для продовження натисніть будь-яку клавішу . . . |
```

## FACTOR 2

### 1. Постановка задачі

2) Скласти програму FACTOR2 – удосконалений варіант програми FACTOR. Ця програма у відповідь на неправильно введене N (тобто не натуральне число) видає зрозумілу підказку користувачеві.

Приклад:

N = -1

Error: N мусить бути додатнім, не negative!

Приклад:

N = 3.14

Error: N must be natural, not real!

Приклад:

N = шість

Error: N must a number, no a string!

*Для перевірки "число/не число" на вході (cin) можна застосувати такий прийом:*

```
cin >> number;
if(!cin)
    ... значить, помилка ...
else
{
    cin >> onechar;
    if(onechar - не "пробел")
        ... значить, помилка...
}
```

### 2. Опис розв'язку + блок-схема

### 3. Вихідний текст програми

```
#include <iostream>
#include <limits>
#include <Windows.h> // Бібліотека мов
using namespace std;
int main()
{
    SetConsoleCP(1251); //для кирилиці
```

```

SetConsoleOutputCP(1251); //для кирилиці
double N, N1, c, dub=1;
cout << "Введи число N";
cin >> N;
if (!cin)
{
    cout << "N must be a number." << endl;
}
else
{
    if (N < 0)
    {
        cout << "ERROR: N must not be negative." << endl;
    }
    if (N == 0)
    {
        dub = 1;
        N1 = N;
        cout << N1 << "! = " << dub << endl;
    }
    if (N == 1)
    {
        dub = 1;
        N1 = N;
        cout << N1 << "! = " << dub << endl;
    }
    else if (N > 0)
    {
        for (double n0 = 1; n0 < N; n0++)

```

```

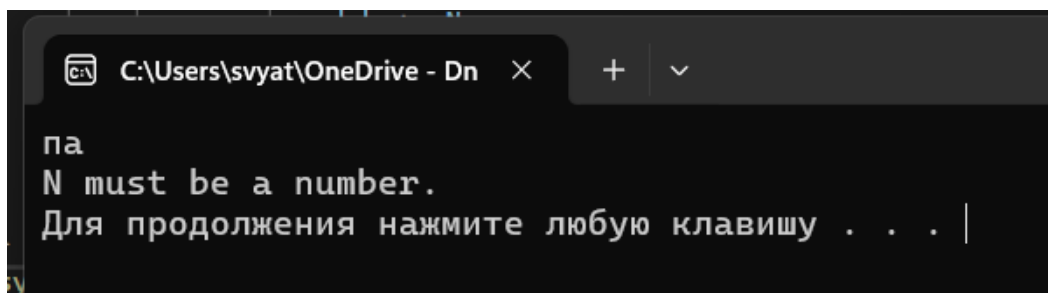
    {
        c = N - n0;
    }
    if (c < 1)
    {
        cout << "ERROR: N must be natural, not real." << endl;
    }
    else
    {
        N1 = N;
        while (N >= 1)
        {
            dub *= N;
            N--;
        }
        cout << N1 << "!" << dub << endl;
    }
}
}
system("pause");
}

```

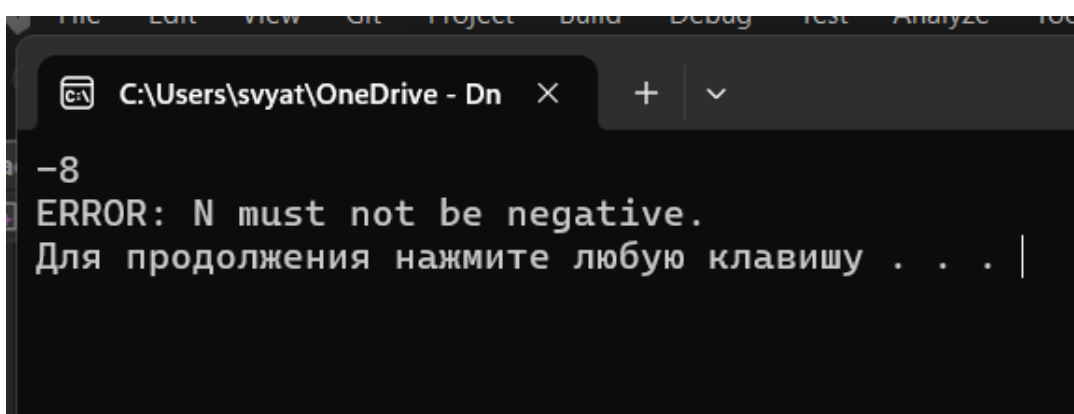
#### 4. Опис інтерфейсу програми

Користувач має запустити файл .exe і вписати число. Він отримає факторіал цього числа (якщо такий факторіал існує). В супротивному випадку він отримає помилку.

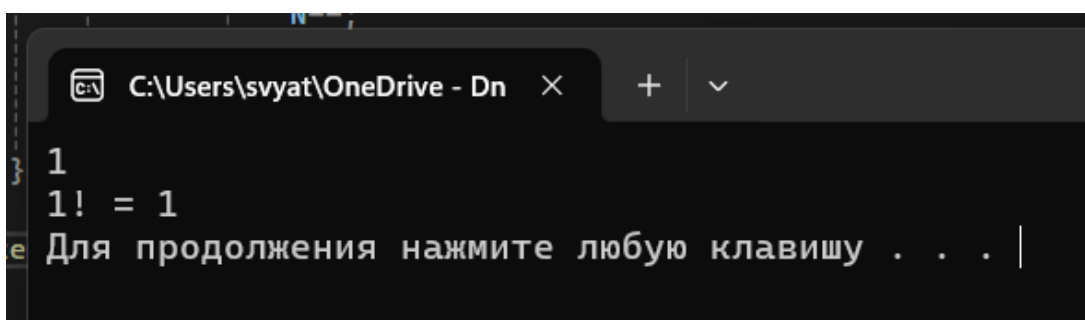
## 5. Опис тестових прикладів



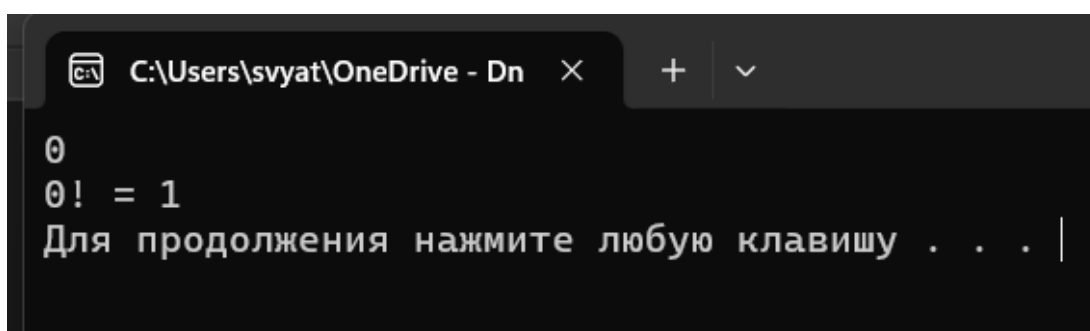
па  
N must be a number.  
Для продолжения нажмите любую клавишу . . . |



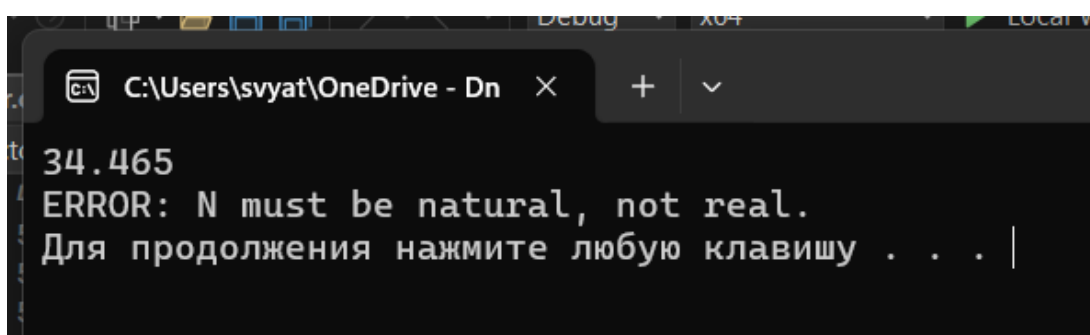
-8  
ERROR: N must not be negative.  
Для продолжения нажмите любую клавишу . . . |



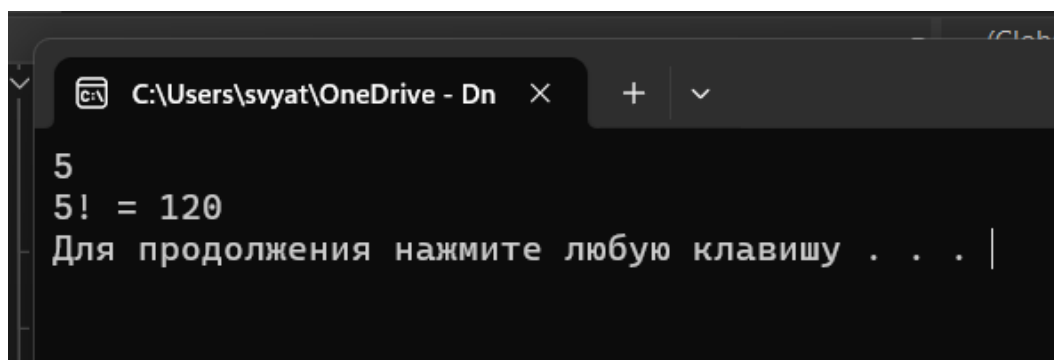
1  
1! = 1  
Для продолжения нажмите любую клавишу . . . |



0  
0! = 1  
Для продолжения нажмите любую клавишу . . . |



34.465  
ERROR: N must be natural, not real.  
Для продолжения нажмите любую клавишу . . . |



A screenshot of a Windows command prompt window. The title bar shows the path "C:\Users\svyat\OneDrive - Dn" and a close button. The command prompt displays the number "5", followed by the calculation "5! = 120". Below this, a Russian instruction "Для продолжения нажмите любую клавишу . . ." is shown, followed by a vertical cursor line.

```
5  
5! = 120  
Для продолжения нажмите любую клавишу . . . |
```

### **Аналіз помилок**

Під час виконання данної лабораторної роботи я часто плутався з циклами, забув крапки з комами, та іноді двічі оголошував змінні.

### **Висновки**

Під час виконання данної лабораторної роботи я значно поглибив власне розуміння циклів. Вони стали для мене набагато простішими. Я навчився використовувати складне присвоєння та ставити точні умови для того, щоб цикл виконувався.



