

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра систем штучного інтелекту



## **Звіт**

про виконання

**VNS Lab1, VNS Lab2, Algotester Lab1, Algotester Lab2**

**з дисципліни:** «Мови та парадигми програмування»

**з розділу:** «Епіс 2. Умовні алгоритми з розгалуженням, оператори керування : if-else, switch, goto, введення в цикли. Константи, змінні»

***Виконав:***

студент групи ШІ-14

Шевченко Данііл Вікторович

## Тема роботи:

Умовні оператори if-else, конструкції типу if – else if – else, умовний оператор switch, введення в цикли for, while, do while, foreach, оператори continue, break, exit

## Мета роботи:

Ознайомитись з алгоритмами розгалуженнями, умовними операторами if-else, конструкціями типу if – else if – else, умовним оператором switch, отримати початкові уявлення про цикли for, while, do while, for each, ознайомитись з операторами управління потоком програми: goto, break, continue, exit, отримати початкові уявлення про стрічний тип std::string, оператор std::getline

Завдання:

1. Epic 2 Task 1 - Theory Education Activities
2. Epic 1 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)
3. Epic 2 Task 3 - Lab# programming: VNS Lab 1 Task 1
4. Epic 2 Task 4 - Lab# programming: VNS Lab 1 Task 2
5. Epic 2 Task 5 - Lab# programming: VNS Lab 2 Task 1
6. Epic 2 Task 6 - Lab# programming: Algotester Lab 1 Task 1
7. Epic 2 Task 7 - Lab# programming: Algotester Lab 2 Task 1
8. Epic 2 Task 8 - Practice# programming: Class Practice Task
9. Epic 2 Task 9 - Practice# programming: Self Practice Task
10. Epic 2 Task 10 - Result Documentation Report and Outcomes Placement Activities (Docs an Programs on GitHub)
11. Epic 2 Task 11 - Results Evaluation and Release

## Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
  - Тема №1: Умовні оператори if, else
  - Тема №2: Оператори керування потоком виконання програми: goto, continue, break, exit
  - Тема №3: Умовний оператор switch
  - Тема №4: Цикли: while, do while, for, foreach
  - Тема №5: Робота зі стрічками std::string
  - Тема №6: Змінні та їх типи
  -
- 2) Індивідуальний план опрацювання теорії:
  - Тема №1: Умовні оператори if, else
    - o Джерела Інформації
      - Практика в НУ “ЛП”

- Курси Ravesli.com
- Що опрацьовано:
  - Умовні оператори if та else
  - Використання умовних операторів у зв'язці if – else if – else
  - Вкладені умовні оператори
- Статус: Ознайомлений
- Початок опрацювання теми: 13.10.23
- Звершення опрацювання теми: 31.10.23
- Тема №2: Оператори керування потоком виконання програми: goto, continue, break, exit
  - Джерела Інформації
    - Практика в НУ “ЛП”
    - Курси Ravesli.com
  - Що опрацьовано:
    - Оператор переходу між частинами коду goto та лейбли-вказівники
    - Оператор continue для дострокового припинення виконання ітерації циклу
    - Оператор break для дострокового припинення виконання всього циклу
    - Оператор exit для закінчення програми з певним цифровим кодом виходу
  - Статус: Ознайомлений
  - Початок опрацювання теми: 13.10.23
  - Звершення опрацювання теми: 31.10.23
- Тема №3: Умовний оператор switch
  - Джерела Інформації
    - Практика в НУ “ЛП”
    - Курси Ravesli.com
  - Що опрацьовано:
    - Використання оператора switch(switch – case) як аналога if - else
    - Використання оператора break у зв'язці з оператором switch
    - Обмеження на використання оператора switch
  - Статус: Ознайомлений
  - Початок опрацювання теми: 13.10.23
  - Звершення опрацювання теми: 31.10.23
- Тема №4: Цикли: while, do while, for, foreach
  - Джерела Інформації
    - Практика в НУ “ЛП”
    - Курси Ravesli.com
  - Що опрацьовано:
    - Цикли while та do while і різниця між ними
    - Використання for як більш короткого аналога до while/do while
    - Модифікований цикл for(foreach) для роботи з масивами
  - Статус: Ознайомлений
  - Початок опрацювання теми: 13.10.23
  - Звершення опрацювання теми: 31.10.23
- Тема №5: Тема №5: Робота зі стрічками std::string
  - Джерела Інформації
    - Практика в НУ “ЛП”
    - Курси Ravesli.com
  - Що опрацьовано:
    - Стрічний тип даних std::string

- Використання масиву елементів символьного типу `char[]` як аналога `std::string`
  - Оператор `std::getline` для коректного зчитування стрічо
- Статус: Ознайомлений
- Початок опрацювання теми: 13.10.23
- Звершення опрацювання теми: 31.10.23
- Тема №6: Змінні та їх типи
- 
- Джерела Інформації
  - Практика в НУ “ЛП”
  - Курси Ravesli.com
- Що опрацьовано:
  - Типи змінних
  - Ініціалізація змінних
- Статус: Ознайомлений
- Початок опрацювання теми: 13.10.23
- Звершення опрацювання теми: 31.10.23
- 

## Виконання роботи:

### VNS LAB1 TASK1

#### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 VNS Lab1 Task 1

- Варіант 24

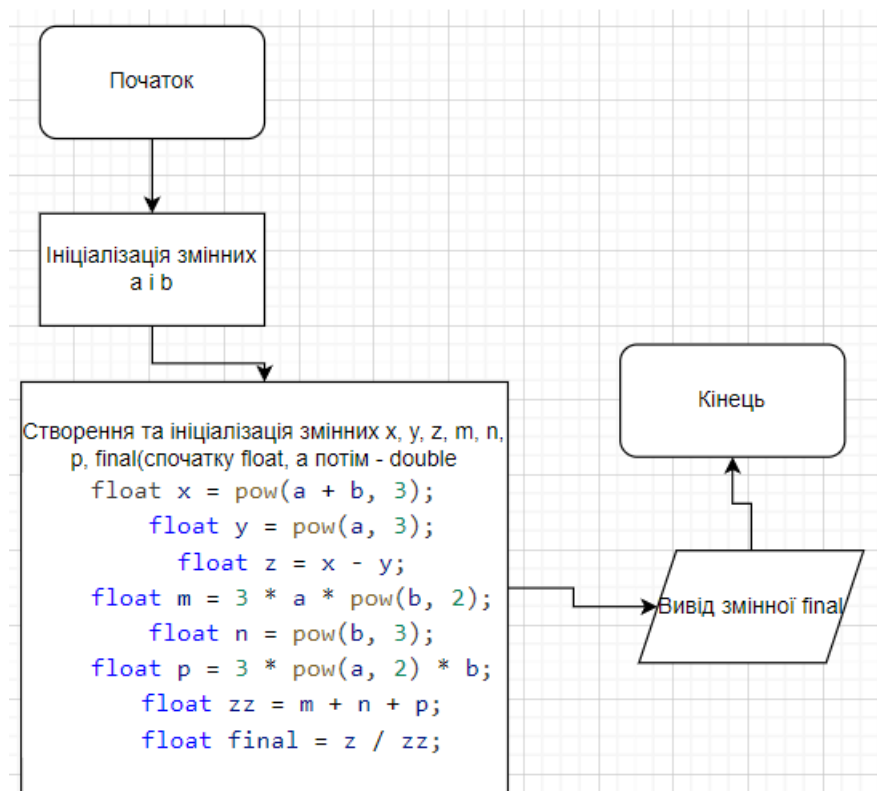
24	$\frac{(a+b)^3 - (a^3)}{3ab^2 + b^3 + 3a^2b},$ <p>при <math>a=1000, b=0.0001</math></p>	1) $n++*m$ 2) $n++<m$ 3) $m-- >m$
----	---	---

- Робота зі змінними та виразами

-

#### 2. Дизайн та планована оцінка часу виконання завдань:

Програма VNS Lab1 Task1



- Планований час на реалізацію – 20 хв
- 

### 3. Код програм з посиланням на зовнішні ресурси:

```

#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    float a = (float)1000;
    float b = 0.0001;
    float x = pow(a + b, 3);
    float y = pow(a, 3);
    float z = x - y;
    float m = 3 * a * pow(b, 2);
    float n = pow(b, 3);
    float p = 3 * pow(a, 2) * b;
    float zz = m + n + p;
    float final = z / zz;
    cout << final << endl;

    double a1 = (double)1000;
    double b1 = 0.0001;
    double x1 = pow(a1 + b1, 3);
    double y1 = pow(a1, 3);
    double z1 = x1 - y1;
    double m1 = 3 * a1 * pow(b1, 2);
    double n1 = pow(b1, 3);
  
```

```

double p1 = 3 * pow(a1, 2) * b1;
double zz1 = m1 + n1 + p1;
double final1 = z1 / zz1;
cout << final1;
}

```

#### 4. Результати виконання завдань, тестування та фактично затрачений час:

```

PS C:\Users\Administrator> & 'c:\Users\Administrator\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\windowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-cobbrgvs.lhe' '--stdout=Microsoft-MIEngine-Out-x1in2ao4.wiu' '--stderr=Microsoft-MIEngine-Error-z1q1bb51.ppl' '--pid=Microsoft-MIEngine-Pid-w01ibuas.5se' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
1.28
1
PS C:\Users\Administrator>

```

Час затрачений на виконання завдання – 20 хв

## VNS LAB1 TASK2

### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №2 VNS Lab1 Task 2

- Варіант 24

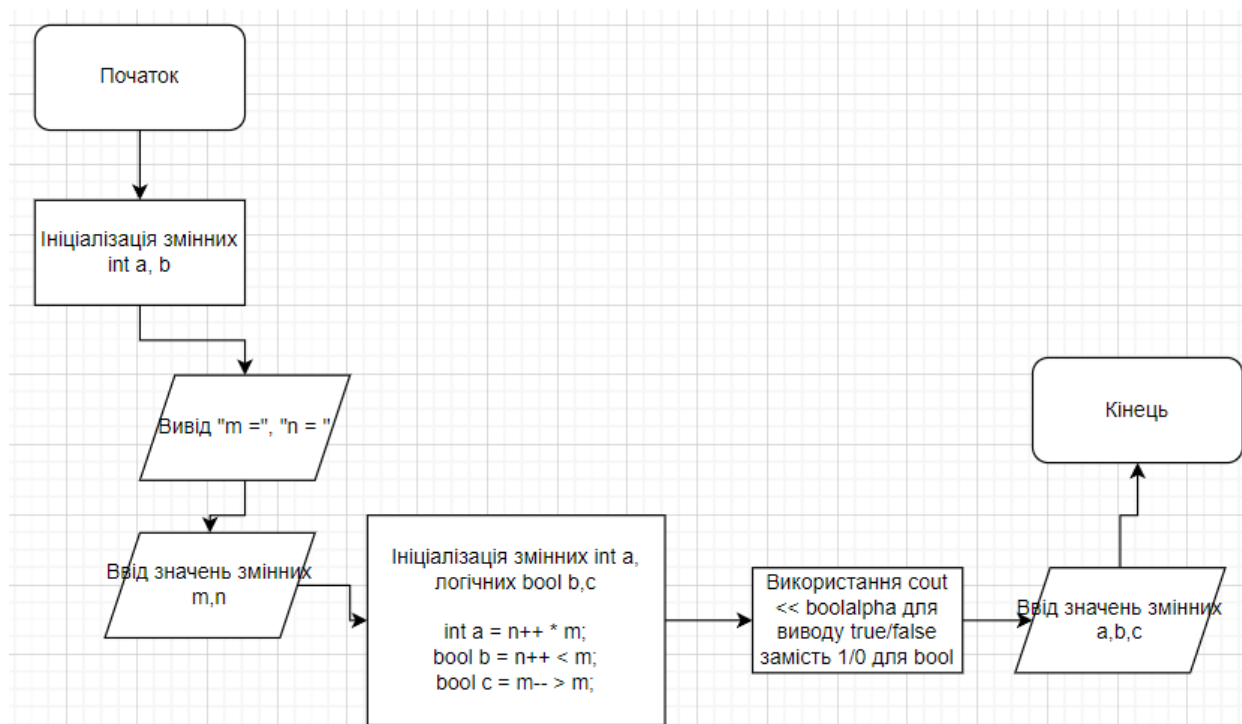
24	$\frac{(a+b)^3 - (a^3)}{3ab^2 + b^3 + 3a^2b},$ <p>при a=1000, b=0.0001</p>	<p>1) n++*m</p> <p>2) n++&lt;m</p> <p>3) m-- &gt;m</p>
----	--	--

- Робота зі змінними та виразами, інкремент та декремент

-

### 2. Дизайн та планована оцінка часу виконання завдань:

Програма VNS Lab1 Task2



- Планований час на реалізацію – 20 хв

### 3. Код програм з посиланням на зовнішні ресурси:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int m;
```

```
    int n;
```

```
    cout << "m = ";
```

```
    cin >> m;
```

```
    cout << "n = ";
```

```
    cin >> n;
```

```
    int a = n++ * m;
```

```
    bool b = n++ < m;
```

```
    bool c = m-- > m;
```

```
    cout << boolalpha;
```

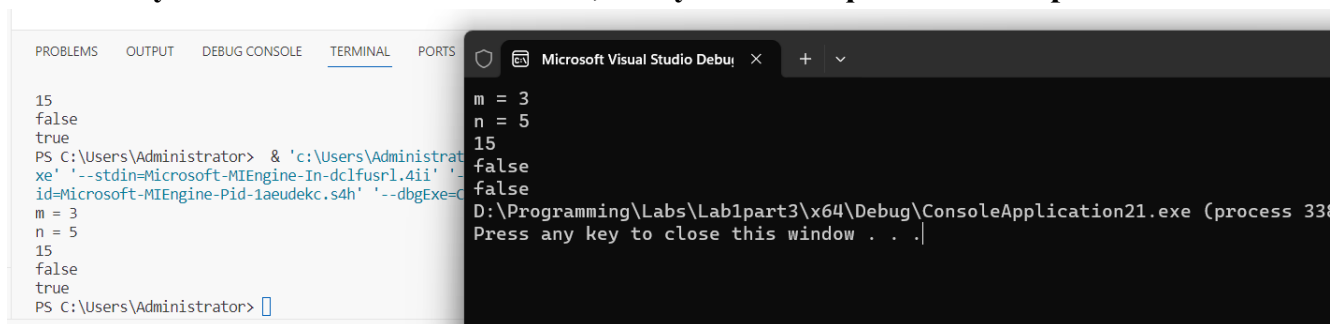
```
    cout << a << endl;
```

```
    cout << b << endl;
```

```
    cout << c;
```

```
}
```

#### 4. Результати виконання завдань, тестування та фактично затрачених час:



Результат виконання програми в VSCode (зліва) та в Visual Studio2022 (справа)

Час затрат на выполнение задания – 20 хв

## VNS LAB2

## 1. Опрацювання завдання та вимог до програм та середовища:

### Завдання №3 VNS Lab2

- ВАРИАНТ 23

23) Знайти суму 10 членів ряду, у якому

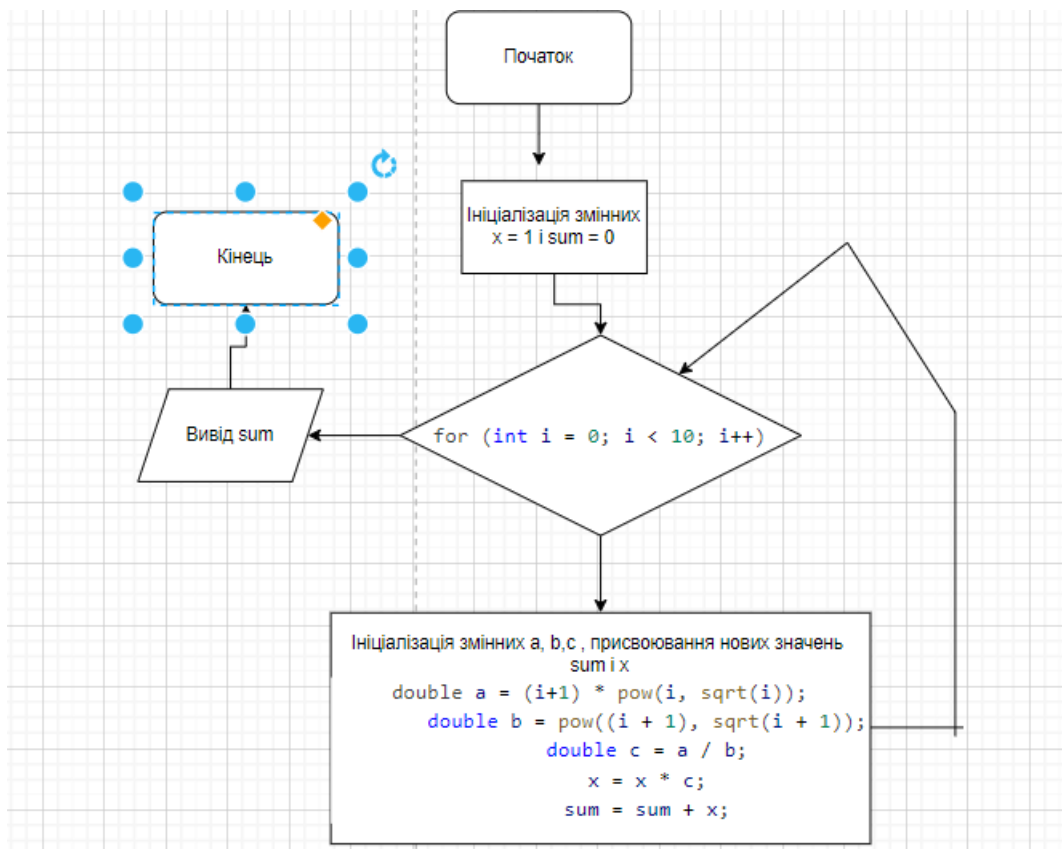
$$a_n = \frac{n!}{n^{\sqrt{n}}}$$

- — —

## 2. Дизайн та планована оцінка часу виконання завдань:

## Програма VNS Lab2





- Планований час на реалізацію – 1 година
- 

### 3. Код програм з посиланням на зовнішні ресурси:

1 спосіб(рекурентна формула):

```

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

int main()
{
    double sum = 0;
    double x = 1;
    for (int i = 0; i < 10; i++)
    {
        double a = (i+1) * pow(i, sqrt(i));
        double b = pow((i + 1), sqrt(i + 1));
        double c = a / b;
        x = x * c;
        sum = sum + x;
    }
    cout << "The sum is " << sum;
}

```

2 спосіб(формула n-го члена):

```

#include <iostream>
#include <math.h>

```

```

using namespace std;
int factorial(int a)
{
    if (a > 0)
    {
        int x = 1;
        for (int i = 1; i <= a; i++)
        {
            x = x * i;
        }
        return x;
    }
    if (a == 0)
    {
        return 1;
    }
}

int main()
{
    double sum = 0;
    for (int n = 1; n <= 10; n++)
    {
        double x = factorial(n) / (pow(n, sqrt(n)));
        sum = sum + x;
    }
    cout << "The sum is " << sum;
}

```

#### 4. Результати виконання завдань, тестування та фактично затрачених час:

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Administrator> & 'c:\Users\Administrator\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-2vugljh.2dv' '--stdout=Microsoft-MIEngine-Out-tnksoi3e.k45' '--stderr=Microsoft-MIEngine-Error-eom2celd.asf' '--pid=Microsoft-MIEngine-Pid-wpqexhlb.ohn' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
The sum is 3153.32
PS C:\Users\Administrator>

```

Результат виконання програми

Час затрачених на виконання завдання – 1.5 години

# Algotester Lab1

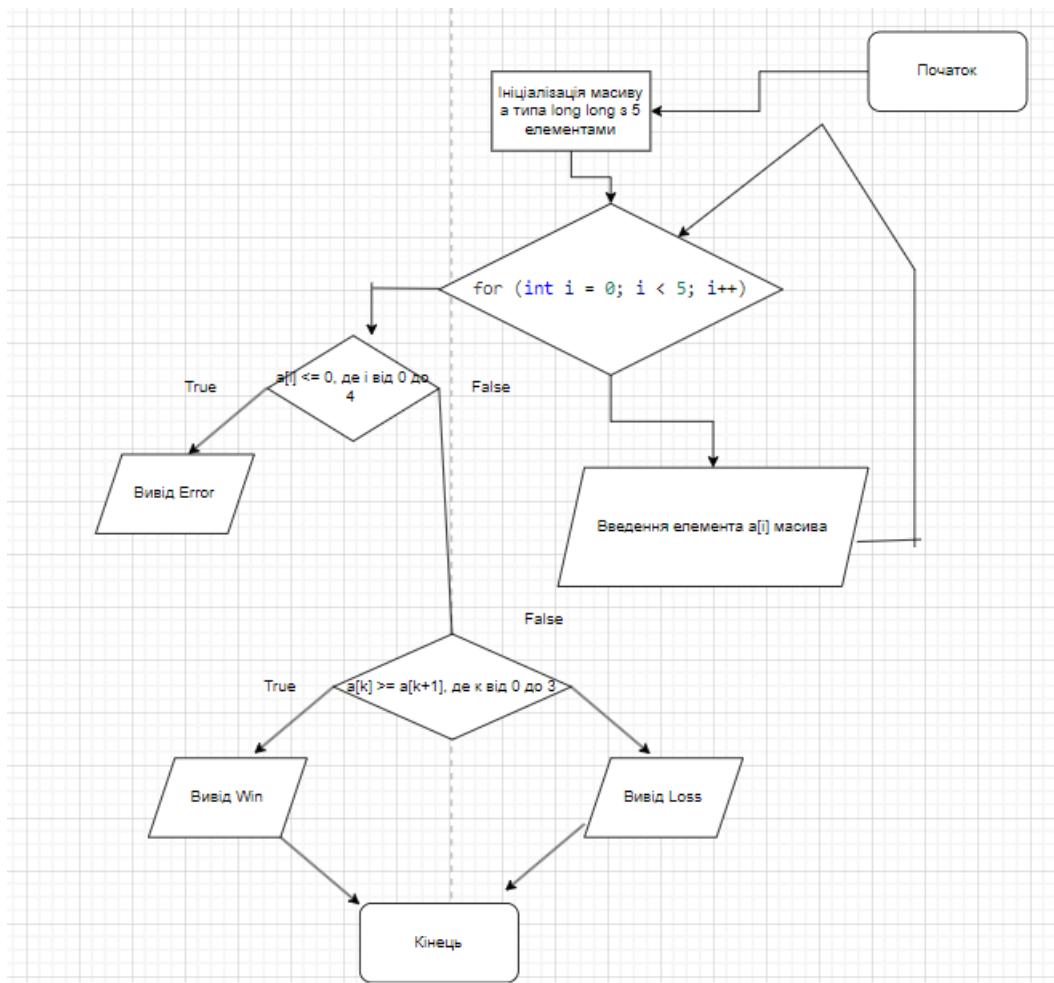
## 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №4 Algotester Lab1

- Варіант 3
- Lab 1v3
- 
- Обмеження: 1 сек., 256 MiB
- Персонажу по одному дають сторони 5 кубів  $a_1..a_5$ , з яких він будує піраміду.
- Коли він отримує куб з ребром  $a_i$  - він його ставить на існуючий, перший ставить на підлогу
- (вона безмежна).
- Якщо в якийсь момент об'єм куба у руці (який будуть ставити) буде більший ніж у куба на вершині піраміди - персонаж програє і гра закінчується. Розмір усіх наступних кубів після
- програшу не враховується.
- Тобто якщо  $a_{i-1} < a_i$  - це програш.
- Ваше завдання - сказати як закінчиться гра.
- Вхідні дані
- 5 цілих чисел  $a_1..a_5$  - сторони кубів
- Вихідні дані
- Існуючі варіанти:
- LOSS - якщо персонаж не зможе поставити куб.
- WIN - якщо персонаж зможе поставити усі куби
- ERROR - якщо сторона куба буде мати неможливу довжину, тобто  $a_i \leq 0$
- Обмеження
- $-1012 \leq a_i \leq 1012$
- 
- 

## 2. Дизайн та планована оцінка часу виконання завдань:

Програма Algotester Lab1



- 
- Планований час на реалізацію – 1 година
- 

### 3. Код програм з посиланням на зовнішні ресурси:

1 спосіб:

```

#include <iostream>
#include <cmath>

int main()
{
    long long a[5];
    for (int i = 0; i < 5; i++)
    {
        std::cout << "Enter the # " << i + 1 << " element: ";
        std::cin >> a[i];
    }
    bool limits = false;
    for (int i = 0; i < 5; i++)
    {
        if (a[i] >= pow(10, 12) || a[i] <= -pow(10, 12))
        {
            break;
        }
    }
}
  
```

```

    }
    if (i == 4)
    {
        limits = true;
    }
}
if (limits)
{
    if (a[0] <= 0 || a[1] <= 0 || a[2] <= 0 || a[3] <= 0 || a[4] <= 0)
    {
        std::cout << "Error";
    }

    else
    {
        if (a[0] >= a[1] && a[1] >= a[2] && a[2] >= a[3] && a[3] >= a[4])
        {
            std::cout << "Win";
        }
        else
        {
            std::cout << "Loss";
        }
    }
}
else
{
    std::cout << "Not in limits";
}
}
2 cnoci6:
#include <iostream>
#include <cmath>

int main()
{
    long long a[5];
    for (int i = 0; i < 5; i++)
    {
        std::cout << "Enter the # " << i + 1 << " element: ";
        std::cin >> a[i];
    }
    bool limits = false;
    for (int i = 0; i < 5; i++)
    {
        if (a[i] >= pow(10, 12) || a[i] <= -pow(10, 12))
        {
            break;

```

```

    }
    if (i == 4)
    {
        limits = true;
    }
}
if (limits)
{
    if (a[0] <= 0 || a[1] <= 0 || a[2] <= 0 || a[3] <= 0 || a[4] <= 0)
    {
        std::cout << "Error";
    }

    else
    {
        bool win = true;
        for (int i = 1; i < 5; i++)
        {
            if (a[i] > a[i-1])
            {
                win = false;
            }

        }

        if (win)
        {
            std::cout << "Win";
        }
        else
        {
            std::cout << "Loss";
        }
    }
}
else
{
    std::cout << "Not in limits";
}
}

```

#### 4. Результати виконання завдань, тестування та фактично затрачений час:

```
PS C:\Users\Administrator> & 'c:\Users\Administrator\.vscode
xe' '--stdin=Microsoft-MIEngine-In-ru1k1szk.ta1' '--stdout=Mi
id=Microsoft-MIEngine-Pid-yvdwlezt.skh' '--dbgExe=C:\msys64\u
Enter the # 1 element: 6
Enter the # 2 element: 4
Enter the # 3 element: 3
Enter the # 4 element: 3
Enter the # 5 element: 1
Win
PS C:\Users\Administrator>
```

```
PS C:\Users\Administrator> & 'c:\Users\Adminis
xe' '--stdin=Microsoft-MIEngine-In-cyjyzqj5.f4b
id=Microsoft-MIEngine-Pid-orc0t2y1.5m5' '--dbgE
Enter the # 1 element: 6
Enter the # 2 element: 4
Enter the # 3 element: 3
Enter the # 4 element: 7
Enter the # 5 element: 2
Loss
PS C:\Users\Administrator>
```

```
Enter the # 5 element: 2
Loss
PS C:\Users\Administrator> & 'c:\Users\Admini
xe' '--stdin=Microsoft-MIEngine-In-muevhxws.qi
id=Microsoft-MIEngine-Pid-3g4xvoqm.cm4' '--dbg
Enter the # 1 element: 6
Enter the # 2 element: -2
Enter the # 3 element: 4
Enter the # 4 element: 6
Enter the # 5 element: 7
Error
PS C:\Users\Administrator>
```

Результат виконання програми

Час затрачений на виконання завдання – 1.5 години

## Algotester Lab2

### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №5 Algotester Lab2

- Варіант 2

Lab 2v2

Обмеження: 1 сек., 256 MiB

У вас є масив  $g$  розміром  $N$ . Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей масив у масив сум, розміром  $N_{\text{new}} - 1$  (розмір нового масиву після видалення елементів), який

буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

Вхідні дані

У першому рядку ціле число  $N$  - кількість чисел

У другому рядку масив  $r$ , який складається з  $N$  цілих чисел

У третьому рядку 3 цілих числа,  $a$ ,  $b$ ,  $c$ , які треба видалити з масиву

Вихідні дані

У першому рядку ціле число  $M$  - кількість чисел у масиві, який буде виведено

У наступному рядку  $M$  чисел - новий масив

Обмеження

$$1 \leq N \leq 105$$

$$0 \leq r_i \leq 105$$

$$0 \leq a, b, c \leq 105$$

-

○

- Планований час на реалізацію – 2 години

-

## 2. Код програм з посиланням на зовнішні ресурси:

1 спосіб

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

int main()
{
    int N;
    cout << "N: ";
    cin >> N;
    vector<int> array;
    array.resize(N);
    int extra[3];
    cout << "Enter " << N << " array elements: " << endl;
    for (int i = 0; i < N; i++)
    {
        cin >> array[i];
    }
    cout << "Enter 3 extra numbers: " << endl;
    for (int i = 0; i < 3; i++)
    {
        cin >> extra[i];
    }

    for (int x = 0; x < N; x++)
```



```

{
    for (int i = 0; i < 3; i++)
    {
        if (array[x] == extra[i])
        {
            array[x] = -1;
        }
    }
}

cout << "The helpful array: " << endl;
for (int i = 0; i < N; i++)
{
    cout << array[i] << " ";
}
cout << endl;

vector<int> newarray;
newarray.resize(N);

int j = 0;
for (int i = 0; i < N; i++)
{
    if (array[i] != -1)
    {
        newarray[j] = array[i];
        j++;
    }
}
newarray.resize(j);
cout << "The new array is: " << endl;

for (int i = 0; i < j; i++)
{
    cout << newarray[i] << " ";
}
cout << endl;

vector<int> finalarray;
finalarray.resize(j - 1);
cout << "The amount of elements in the final array is: " << j - 1 << endl;
cout << "The final array is: " << endl;
for (int i = 0; i < j - 1; i++)
{
    finalarray[i] = newarray[i] + newarray[i + 1];
    cout << finalarray[i] << " ";
}
}

```

```

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

int main()
{
    int N;
    cout << "N: ";
    cin >> N;
    vector<int> array;
    array.resize(N);
    int extra[3];
    cout << "Enter " << N << " array elements: " << endl;
    for (int i = 0; i < N; i++)
    {
        cin >> array[i];
    }
    cout << "Enter 3 extra numbers: " << endl;
    for (int i = 0; i < 3; i++)
    {
        cin >> extra[i];
    }

    vector<int> newarray;
    newarray.resize(N);

    int j = 0;
    for (int i = 0; i < N; i++)
    {
        if (array[i] != extra[0] && array[i] != extra[1] && array[i] != extra[2])
        {
            newarray[j] = array[i];
            j++;
        }
    }
    newarray.resize(j);
    cout << "The new array is: " << endl;

    for (int i = 0; i < j; i++)
    {
        cout << newarray[i] << " ";
    }
    cout << endl;

    vector<int> finalarray;
    if (j > 1)
    {
        finalarray.resize(j - 1);
    }
}

```

```

    cout << "The amount of elements in the final array is: " << j - 1 << endl;
    cout << "The final array is: " << endl;
    for (int i = 0; i < j - 1; i++)
    {
        finalarray[i] = newarray[i] + newarray[i + 1];
        cout << finalarray[i] << " ";
    }
}
else
{
    cout << "Final result: 0";
}
}

```

### 3. Результати виконання завдань, тестування та фактично затрачених час:

```

N: 6
Enter 6 array elements:
1 2 3 4 5 7
Enter 3 extra numbers:
4 5 6
The helpful array:
1 2 3 -1 -1 7
The new array is:
1 2 3 7
The amount of elements in the final array is: 3
The final array is:
3 5 10
PS C:\Users\Administrator>

```

```

N: 3
Enter 3 array elements:
1 2 3
Enter 3 extra numbers:
1 2 3
The helpful array:
-1 -1 -1
The new array is:

Final result: 0
PS C:\Users\Administrator>

```

```
N: 4
Enter 4 array elements:
1 2 3 4
Enter 3 extra numbers:
1 2 3
The helpful array:
-1 -1 -1 4
The new array is:
4
Final result: 0
PS C:\Users\Administrator>
```

Результат виконання програми

Час затрачений на виконання завдання – 4 години

## Practic 2

### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №6 Practic 2

Еріс 2 Practice - Оператори галуження

Ви створюєте простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди.

Вимоги

Запропонувати користувачеві ввести поточні погодні умови.

Варіанти

- sunny;
- rainy;
- cloudy;
- snowy;
- windy;

Якщо користувач вводить будь-яку іншу умову, запропонуйте йому ввести дійсну умову.

Використовуйте таку логіку

if else - щоб вирішити, чи повинен користувач взяти куртку чи ні.

if, else if - щоб надати рекомендацію щодо активності (прогулянка, футбол, настільні ігри, etc).

switch case - для визначення типу рекомендованого взуття.

Деталі логіки

Рішення чи брати куртку (використовуючи if else)

Якщо йде сніг або дощ, користувач повинен одягнути куртку.

В іншому випадку куртка не потрібна.

Рекомендація щодо активності (використання if, else if)

Якщо сонячно, порекомендуйте «Чудовий день для пікніка!».

Інакше, якщо буде дощ, рекомендуємо «Ідеальна погода, щоб читати книгу всередині!».

Інакше, якщо хмарно, рекомендуємо «Може, відвідати музей?».

Інакше, якщо сніг, порекомендуйте «Як щодо того, щоб зліпити сніговика?».

Інакше, якщо буде вітер, порекомендуйте «Запустіть повітряного змія, якщо він у вас є!».

Рекомендації щодо взуття (з використанням футляра для вимикача)

sunny -> "Взуй улюблені кросівки!"

rainy -> "Дощові чоботи - гарна ідея!"

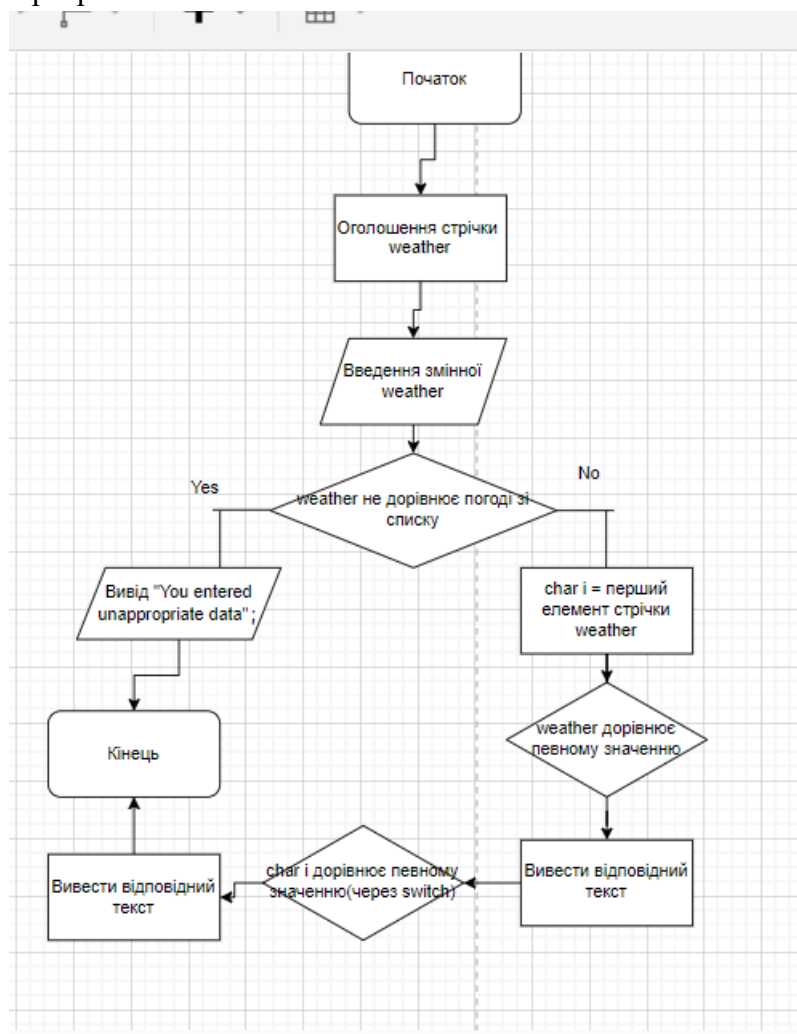
cloudy -> "Сьогодні підходить будь-яке взуття."

snowy -> "Снігові черевики зігріють ваші ноги!"

windy -> "Одягніть щось міцне!"

## 2. Дизайн та планована оцінка часу виконання завдань:

Програма Practic2



- Планований час на реалізацію – 30 хвилин

-

### 3. Код програм з посиланням на зовнішні ресурси:

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

int main()
{
    string weather;

    cout << "Enter the weather conditions: ";
    getline(std::cin, weather);
    if (weather != "cloudy" && weather != "snowy" && weather != "rainy" && weather !=
"windy" && weather != "sunny")
    {
        cout << "You entered inappropriate data";
    }

    else
    {
        char i = weather[0];
        if (weather == "snowy" || weather == "rainy")
        {
            cout << "You must take the jacket" << endl;;
        }
        else
        {
            cout << "You may not take the jacket" << endl;;
        }

        if (weather == "sunny")
        {
            cout << "Nice day for a picnic!" << endl;
        }
        else if (weather == "rainy")
        {
            cout << "An ideal for reading a book" << endl;
        }
        else if (weather == "cloudy")
        {
            cout << "Maybe, visit the museum?" << endl;
        }
        else if (weather == "snowy")
        {
            cout << "What about making snowman?" << endl;
        }
        else
        {
            cout << "What about flying a kite?" << endl;
        }
    }

    switch(i)
```

```

{
case 's':
    if (weather == "sunny")
    {
        cout << "Wear your favourite shoes" << endl;
    }
    else
    {
        cout << "Snow boots will warm up your foots" << endl;
    }
    break;
case 'r':

    cout << "Rainy boots is an ideal shoes" << endl;
    break;
case 'c':
    cout << "You can wear any shoes" << endl;
    break;
case 'w':
    cout << "Wear something strong" << endl;
    break;

}

}
}

```

#### 4. Результати виконання завдань, тестування та фактично затрачений час:

```

Enter the weather conditions: rainy
You must take the jacket
An ideal for reading a book
Rainy boots is an ideal shoes
PS C:\Users\Administrator>

```

```

Enter the weather conditions: windy
You may not take the jacket
What about flying a kite?
Wear something strong
PS C:\Users\Administrator>

```

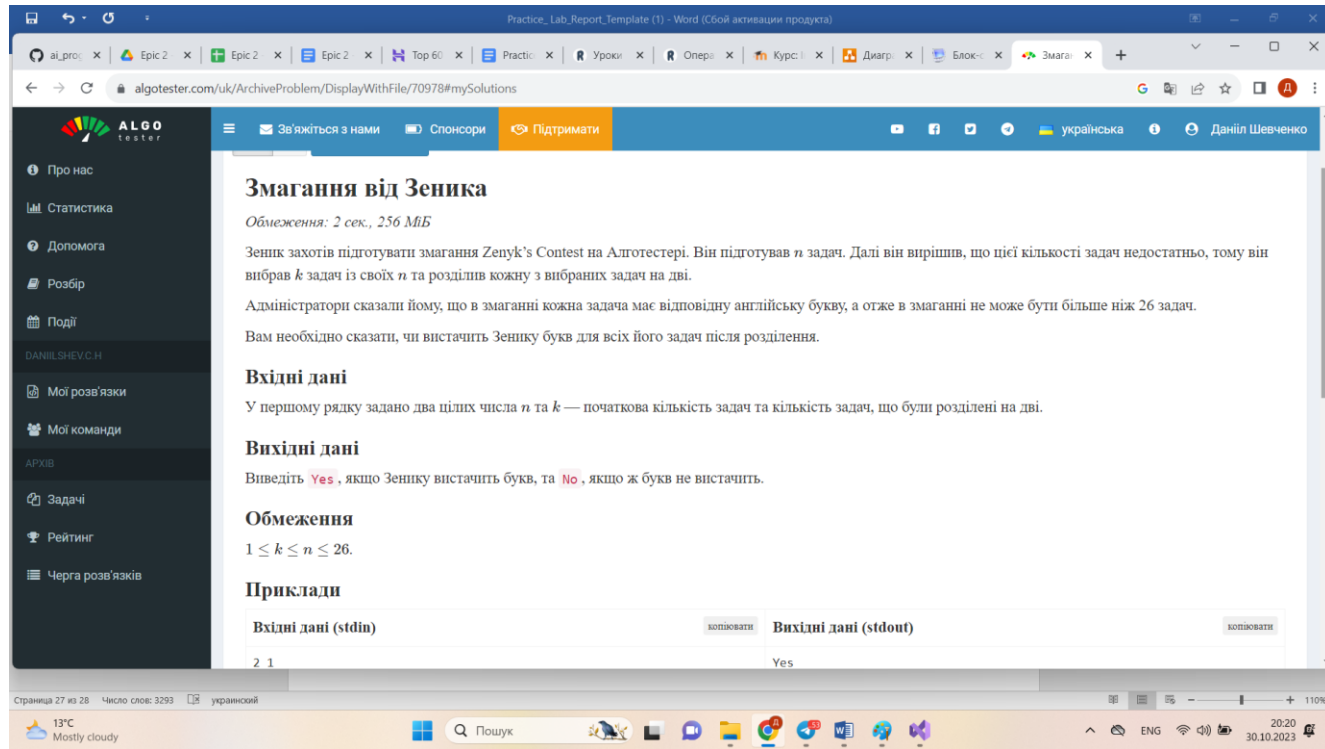
Результат виконання програми

Час затрачений на виконання завдання – 45 хвилин

## Self-Practice

### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №7 Self-practice

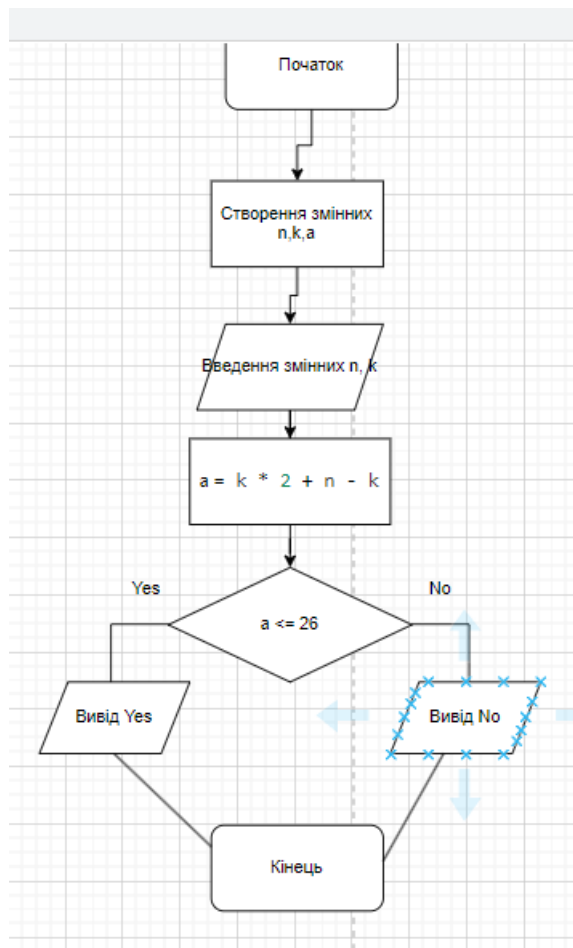


### 2. Дизайн та планована оцінка часу виконання завдань:

Програма Self-Practice

- 
- Планований час на реалізацію – 30 хвилин





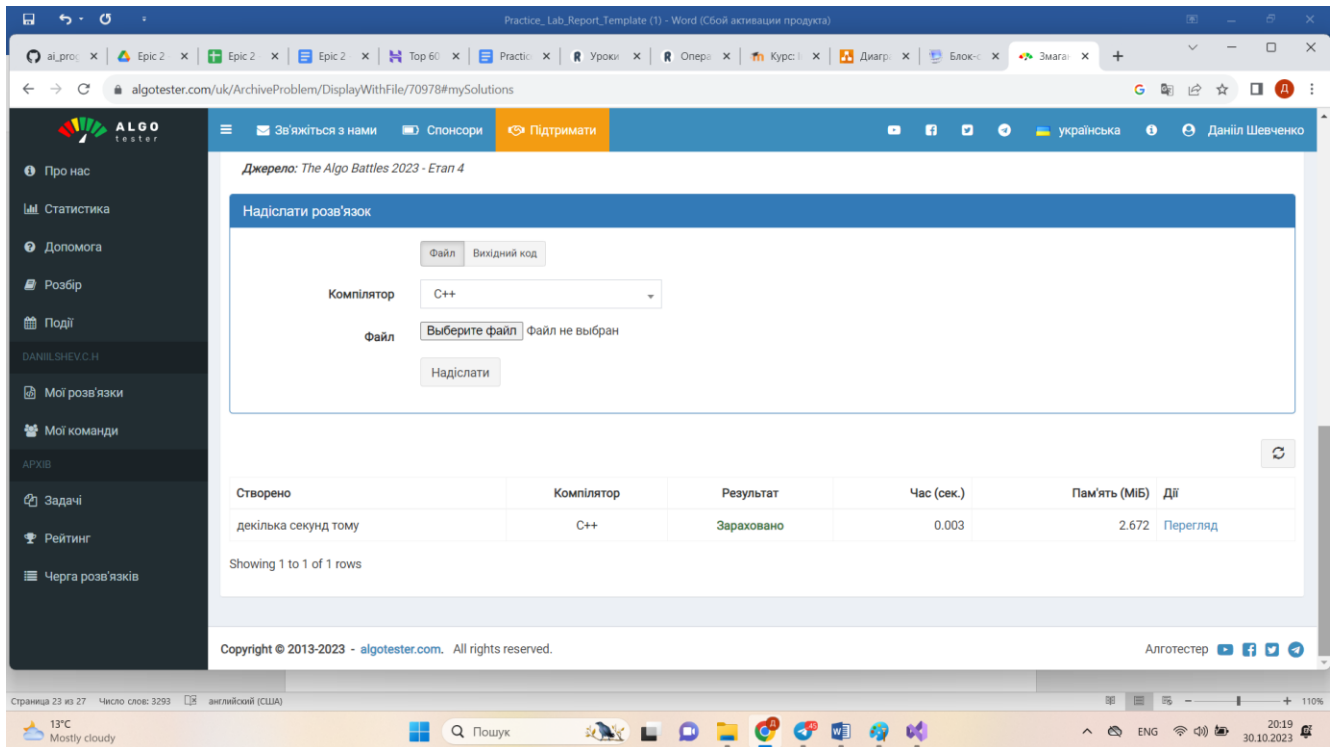
### 3. Код програм з посиланням на зовнішні ресурси:

```

#include <iostream>
using namespace std;
int main()
{
    int n;
    int k;
    int a;
    cin >> n >> k;
    a = k * 2 + n - k;
    if (a <= 26)
    {
        cout << "Yes";
    }
    else
    {
        cout << "No";
    }
}

```

#### 4. Результати виконання завдань, тестування та фактично затрачений час:



Результат виконання програми

Час затрачений на виконання завдання – 45 хвилин

### Practic 3

#### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №8 Practic 3

#### 2. Дизайн та планована оцінка часу виконання завдань:

Програма Practic 3

Еріс 3 Practice - Цикли

Task Description

## Менеджмент бібліотеки

Ви створюєте просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

Програма повинна вміти

- Перерахувати всі книги.
- Дозволити взяти книгу (за наявності).
- Дозволити повернення книги.

Структури даних

- Використовуйте масив або вектор для зберігання назв книг.
- Використовуйте інший масив або вектор для збереження стану доступності кожної книги.

Вимоги

- while: продовжувати працювати, доки користувач не вирішить вийти.
- do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
- for: список усіх книг за допомогою циклу for.
- для кожної: перевірити наявність кожної книги.
- goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

○

- Планований час на реалізацію – 3 години
- 

## 2. Код програм з посиланням на зовнішні ресурси:

```
#include <iostream>
#include <string>
#include <vector>
#include <windows.h>
using namespace std;

int main()
{
    vector<string> books;
    books = { "50 shapes of grey", "Harry Potter", "Zakhar Berkut", "Kobzar",
"Faust", "1984", "Vacation" };
    vector<string> available;
    available = books;
    int size = books.size();
    int choice = 0;
```

```

while (true)
{
tryagain0:
    cout << "1 - list the books\n2 - borrow a book\n3 - return a book\n4 - check
avaibility of a book\n5 - exit" << endl;
    cin >> choice;

    if (choice == 1)
    {
        cout << "\033[35m" << "All existing books:" << "\033[37m" << endl;
        for (int i = 0; i < size; i++)
        {
            cout << books[i] << endl;
        }
        cout << endl;
        cout << "\033[35m" << "Available books:" << "\033[37m" << endl;
        for (string x : available)
        {
            if (x != "none")
            {
                cout << "\033[32m" << x << "\033[37m" << endl;
            }
        }
        cout << endl;
    }

    if (choice == 2)
    {
        cin.ignore(30000, '\n');
        string strchoice;
        do
        {
            cout << "Enter the name of a book or type <<return>> to quit: ";
            getline(cin, strchoice);

            bool ok = false;
            for (int i = 0; i < size; i++)
            {
                if (strchoice == available[i])
                {
                    available[i] = "none";
                    ok = true;
                }
            }
            if (ok)
            {

```

```

        cout << "\033[32m" << "You successfully borrowed a book" <<
"\033[37m" << endl;
    }
    if (!ok && strchoice != "return")
    {
        cout << "\033[31m" << "There is no such book. Try again" <<
"\033[37m" << endl;
    }
    } while (strchoice != "return");
}
if (choice == 3)
{
    cin.ignore(30000, '\n');
    string strchoice2;
    do
    {
        cout << "Enter the name of a book you want to return or type
<<return>> to quit: ";
        getline(cin, strchoice2);
        bool ok2 = false;
        for (int i = 0; i < size; i++)
        {
            if (strchoice2 == books[i] && available[i] == "none")
            {
                available[i] = strchoice2;
                ok2 = true;
            }
        }
        if (ok2)
        {
            cout << "\033[32m" << "You successfully returned the book" <<
"\033[37m" << endl;
        }
        if (!ok2 && strchoice2 != "return")
        {
            cout << "\033[31m" << "The book doesn't belong to our library or
is already available" << "\033[37m" << endl;
        }
    } while (strchoice2 != "return");
}
if (choice == 4)
{
    cin.ignore(30000, '\n');
    string strchoice3;
    do
    {
        cout << "Enter the name of a book you want to check on availability: ";
        getline(cin, strchoice3);
    }

```

```

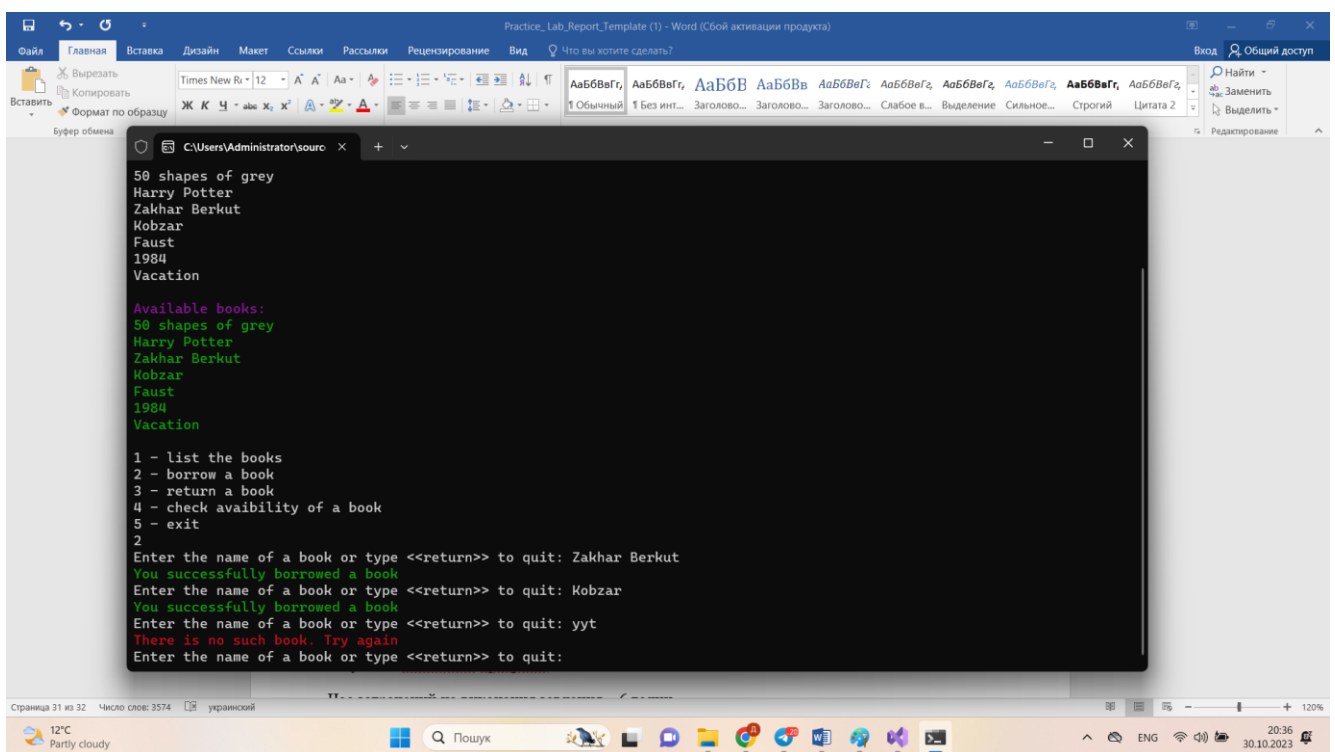
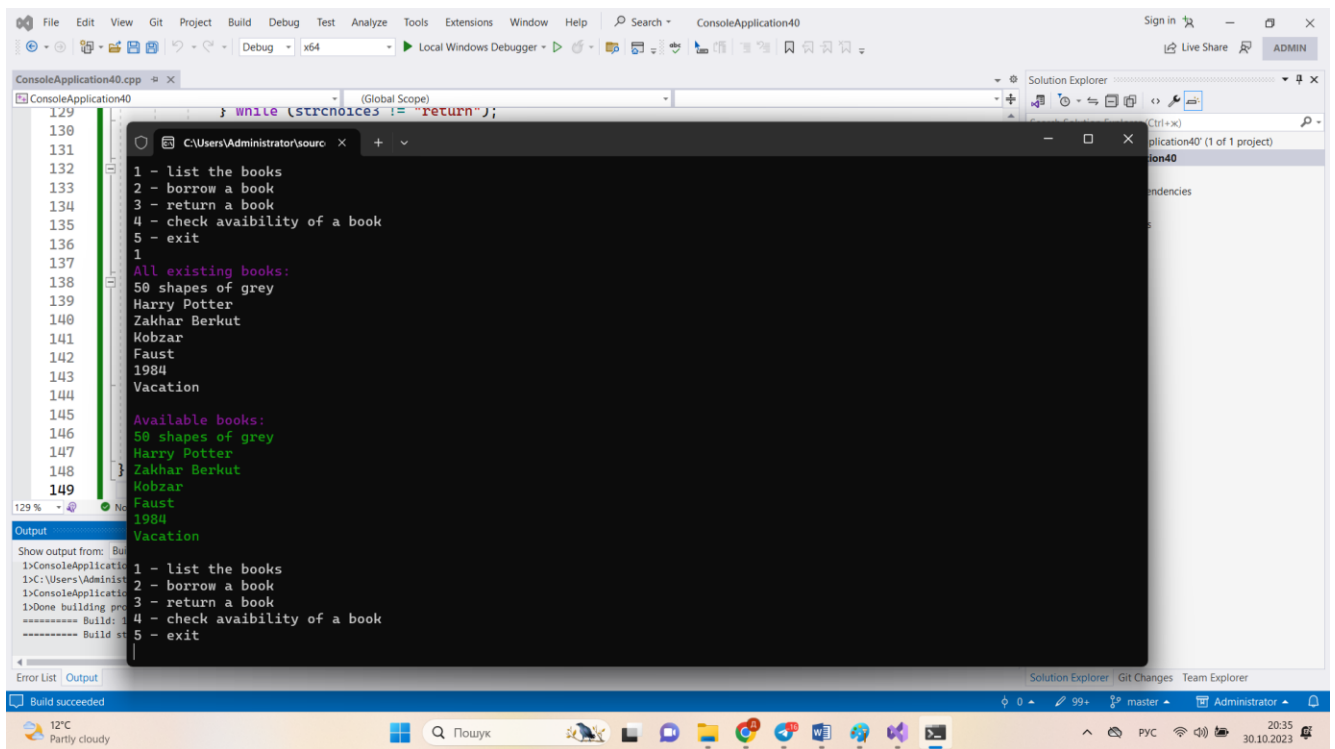
        bool ok3 = false;
        for (string y : available)
        {
            if (y == strchoice3)
            {
                ok3 = true;
            }
        }
        if (ok3)
        {
            cout << "\033[32m" << "The book is available" "\033[37m";
        }
        if (!ok3 && strchoice3!="return")
        {
            cout << "\033[31m" << "The book isn't available" << "\033[37m";
        }
        cout << endl;
    } while (strchoice3 != "return");

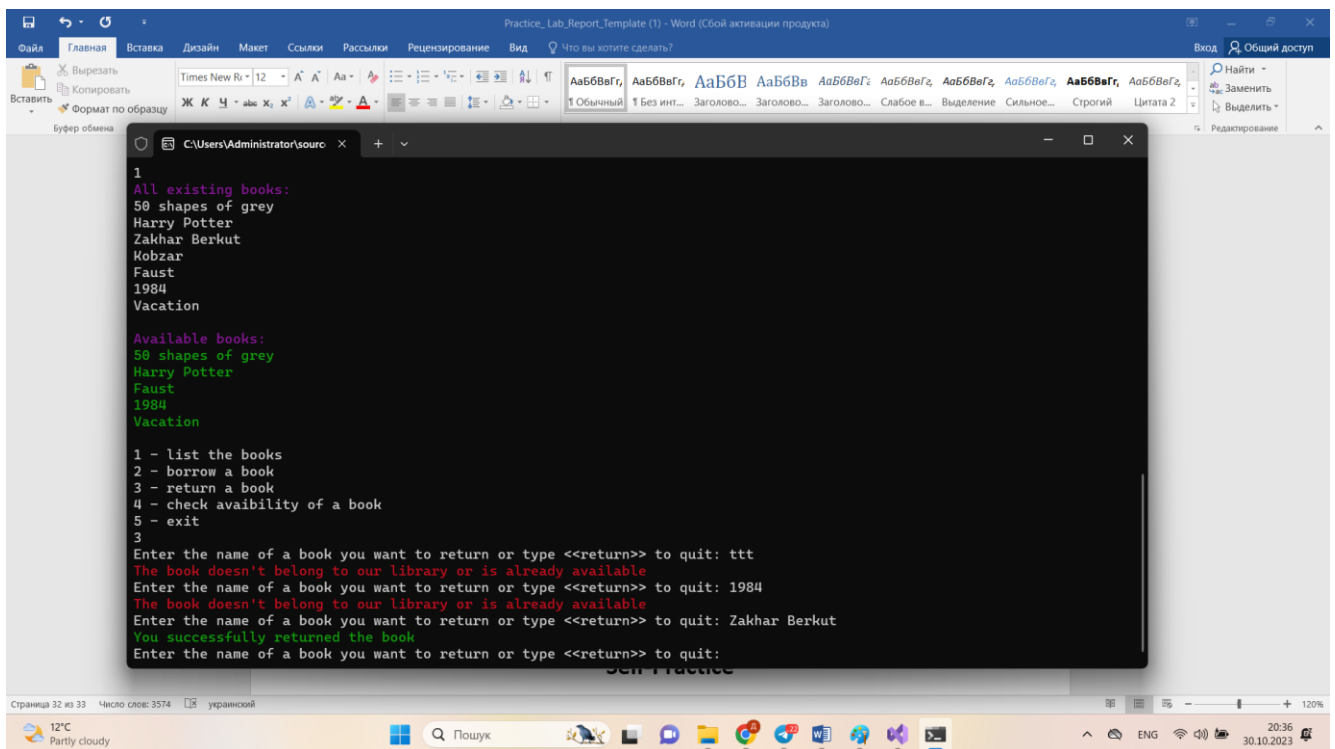
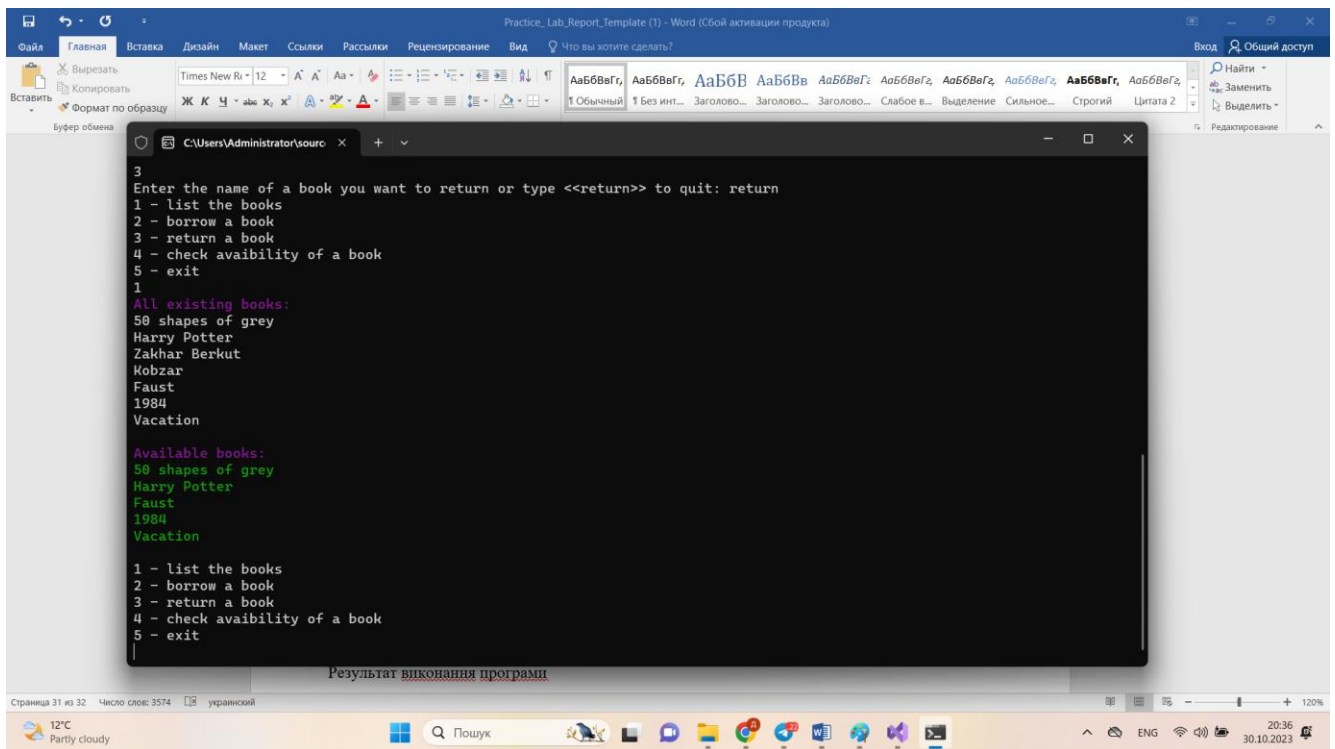
}
if (choice == 5)
{
    cout << "\033[36m" << "Powered by Daniil Shevchenko. Goodbye" <<
"\033[37m";
    Sleep(3000);
    break;
}
if (cin.fail())
{
    cin.clear();
    cin.ignore(30000, '\n');
    goto tryagain0;
}

}
}

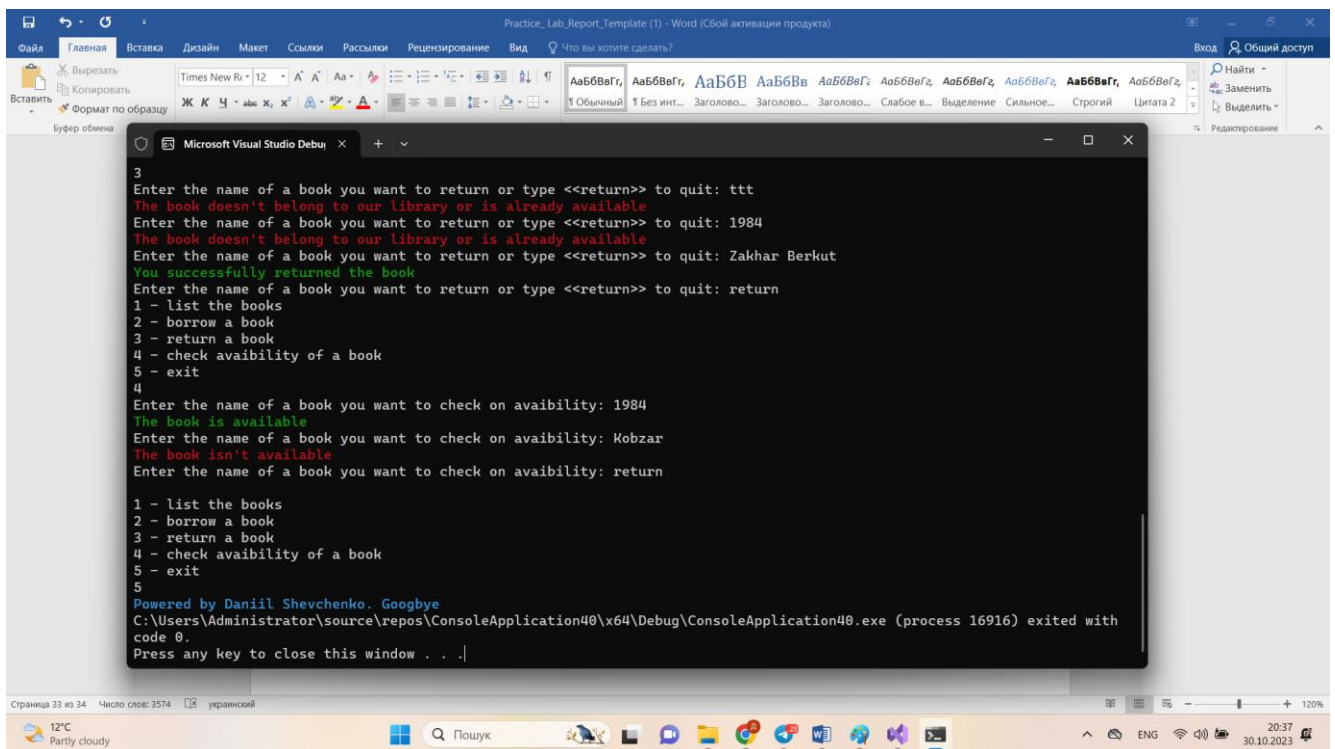
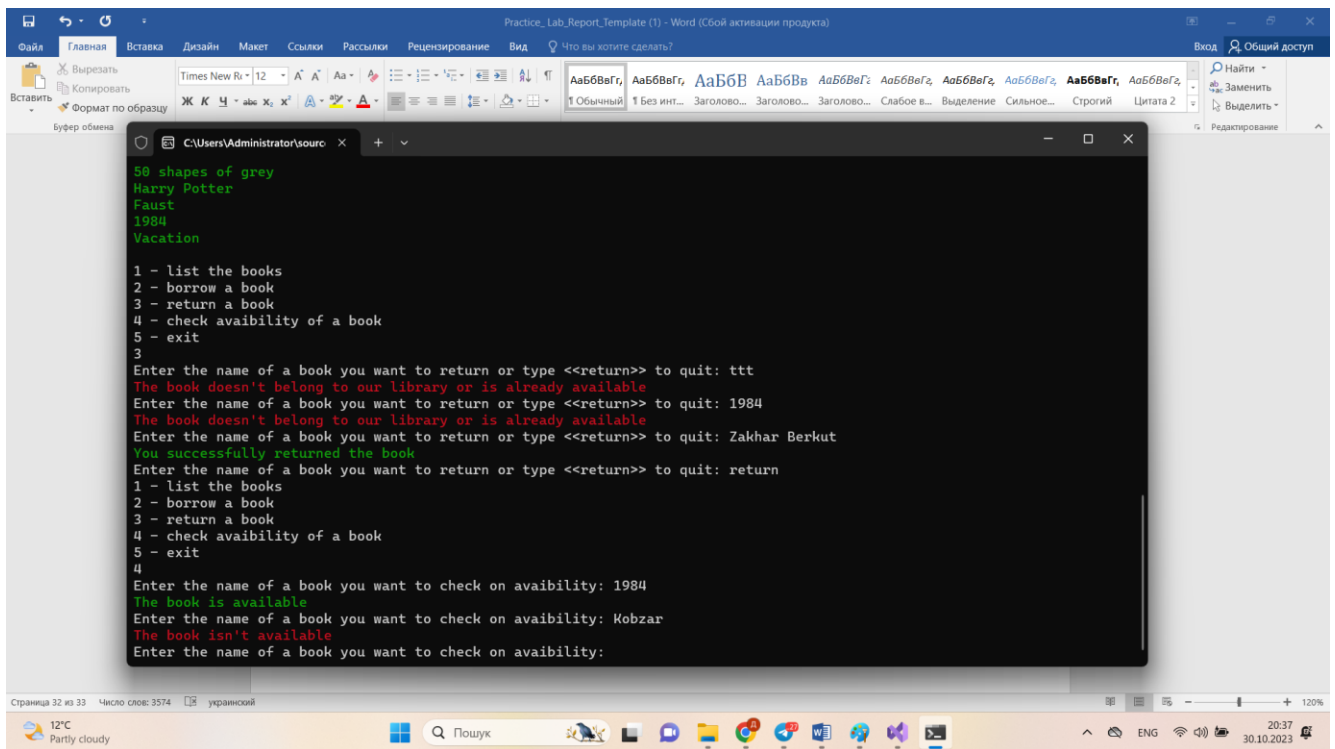
```

### 3. Результати виконання завдань, тестування та фактично затрачених час:









Результат виконання програми

Час затрачений на виконання завдання – 6 годин

## **Висновки:**

Отже, в ході роботи над епіком 2, зокрема 2 ВНС Лабараторними, 2 задачами з Алготестер та 2 практичними роботами, ми вивчили такі понятті мови C++, як алгоритми розгалуження(if-else, switch), цикли(for, while, do while, foreach), оператори управління потоком виконання програми(break, continue, exit, goto). Також ми частково ознайомились з поняттями типу даних string та vector.

Посилання на пулреквест: [https://github.com/artificial-intelligence-department/ai\\_programming\\_playground/pull/182](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/182)