

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



## **Звіт**

### **про виконання лабораторних та практичних робіт блоку № 1**

На тему: «Програмування: алгоритм, програма, код. Системи числення.  
Двійкова система числення. Розробка та середовище розробки програми.»

**з дисципліни: «Основи програмування»**

до:

Практичних Робіт до блоку № 1

Виконала:

Студентка групи ШІ-13

Ходацька Аліна Віталіївна

Львів 2024

**Тема:**

Налаштування IDE (Visual Studio Code). Встановлення та налаштування Git. Синхронізація Git з Github. Вивчення основ C++. Вивчення базових консольних команд Linux. Система числення.

**Мета:**

Навчитися орієнтуватися в своєму робочу середовищі (Visual Studio Code). Встановити компілятор C++ і його налаштувати. Встановити Git, створити SSH ключ і підключитись до GitHub, вивчити команди Git. Ознайомитись з консольними командами Linux та вміти застосовувати їх на практиці. Створити групу команди та дошку завдань, провести онлайн-зустрічі.

**Теоретичні відомості:****1. Вивчила:**

- а) Системи числення, переведення чисел в різні системи.
- б) Базові консольні команди Linux.
- в) Основи мови C++
- г) Git та його конфігурація

**2. Індивідуальний план опрацювання теорії:**

Тема: Системи Числення. Арифметичні дії у двійковій системі числення.

- а) <https://youtu.be/1gJSVxylvQY?si=M5EBIKRHBcZKgcyl>
- б) <https://youtu.be/C5EkxfNEMjE?si=H4ipZrvA951RVynf>

Тема: Linux console commands.

- с) <https://youtu.be/I4EWvMFj37g?si=wr-CzsT4sFals88u>
- д) <https://youtu.be/s3ii48qYBxA?si=cnhEY6s0EXtIYtmt>

Тема: C++

- е) <https://youtu.be/ZzaPdXTrSb8?si=ZR7DVS9P6duwmmY0>
- ф) [https://youtu.be/RSDzvIXmQi4?si=VNDII8TPs1GN\\_L3n](https://youtu.be/RSDzvIXmQi4?si=VNDII8TPs1GN_L3n)
- г) <https://youtu.be/McojvctVsUs?si=C2HbITAYE3GMCY8H>

Тема: Git

- h) <https://youtu.be/4Ze7HAcsuJY?si=B3HRKdPHNjAxU365>
- і) <https://youtu.be/8JJ101D3knE?si=DNf7PHaFm8yIwfxB>
- j) <https://youtu.be/8JJ101D3knE?si=DNf7PHaFm8yIwfxB>

## Виконання роботи:

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

**Завдання № 1** (Class Practice Work. Обчислення складних відсотків за депозитом)

( practice\_work\_task\_1\_alina\_khodatska.cpp )

#### Умова:

Обчислити складні відсотки для депозиту, який був відкритий в банку на певний період часу під фіксовані відсотки з різними варіантами виплати відсотків.

Виплати можливі:

- кожного місяця
- кожного кварталу
- кожного року

#### Вимоги:

- Використати функції `scanf` та `printf` для зчитування і форматування вводу/виводу;
- В кінці програма має вивести повну інформацію про вкладені кошти, загальну суму інвестиції і суму самого заробітку.

**Завдання № 2** (Self Practice Work. Algotester “A+B”)  
(self\_practice\_work\_algotester\_task\_1\_alina\_khodatska.cpp )

Умова:

Дано два цілих числа  $a$  та  $b$ . Завдання — обчислити їхню суму.

Вимоги:

- У єдиному рядку задано два цілих числа  $a$  та  $b$ , які треба додати.
- У єдиному рядку виведіть одне число — суму  $a$  та  $b$ .
- $0 \leq a, b \leq 100$

**Завдання №3** (Self Practice Work. Algotester “Найбільша зростаюча підпоследовність”)

(self\_practice\_work\_algotester\_task\_2\_alina\_khodatska.cpp )

Умова: Задано последовність із  $n$  цілих чисел  $a_j$ . Задача — знайти довжину найбільшої зростаючої підпоследовності заданої последовності.

Вимоги:

- У першому рядку задано ціле число  $n$ .
- У наступному рядку задано  $n$  цілих чисел — последовність  $a_j$ .
- Виведіть довжину найбільшої зростаючої підпоследовності.

**Завдання №4** (Self Practice Work. Algotester “ Офісна Вулиця. Частина 1”)  
(self\_practice\_work\_algotester\_task\_3\_alina\_khodatska.cpp )

Умова: Зустрілися якось працівники великих компаній і почали... Обговорювати план вулиці. Виявляється, всі приміщення, які орендуватимуть ці компанії, збудують вздовж однієї вулиці.  $i$ -та компанія орендуватиме офіс довжиною  $l_i$  метрів. Офіси будуватимуть один за одним, починаючи з точки 0. Всі працівники приїжджатимуть на стоянку, яку побудують в точці 0, та будуть йти до офісів своїх компаній. Тобто, якщо офіси будуть збудовані в порядку  $p_1, p_2, \dots, p_n$ , то перший офіс почнеться в точці 0 і закінчиться в точці  $l_{p_1}$ , другий почнеться в  $l_{p_1}$  і закінчиться в  $l_{p_1} + l_{p_2}$  і т.д. Двері кожного офісу завжди є в кінці будинку, який є ближчим до стоянки. Завдання — допомогти розмістити офіси компаній на цій вулиці в такому порядку, щоб сумарна відстань від точки 0 до усіх офісів була мінімальною.

Вимоги:

- У першому рядку задане ціле число  $n$  — кількість компаній.
- У наступному рядку задано  $n$  цілих чисел  $l_i$  через пробіл — довжини офісів усіх компаній.
- У єдиному рядку виведіть  $n$  чисел від 1 до  $n$  — порядок компаній, в якому варто будувати офіси.
- Якщо існує декілька оптимальних порядків — виведіть будь-який із них.

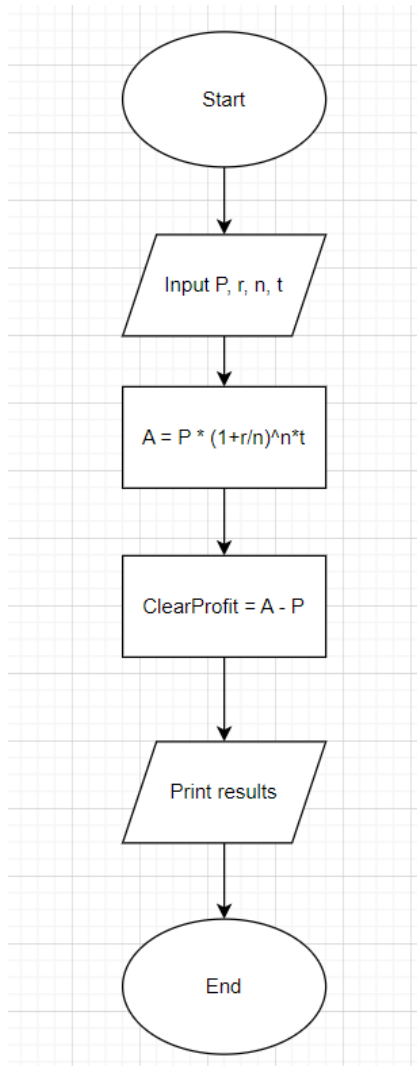
2. Дизайн та планована оцінка часу виконання завдань:

**Завдання №1** (Class Practice Work. Обчислення складних відсотків за депозитом)

Час виконання

Плановий: 1 година

Реальний: 1,5 години

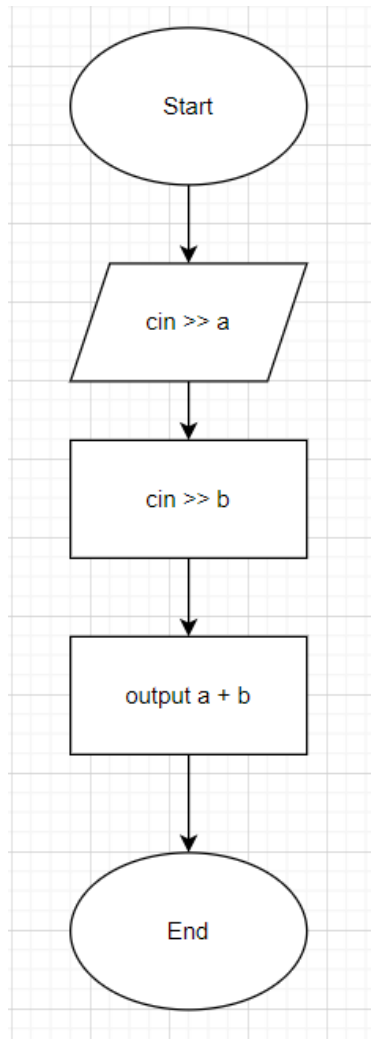


## Завдання № 2 (Self Practice Work. Algotester “A+B”)

Час виконання

Плановий: 20 хвилин

Реальний: 15 хвилин



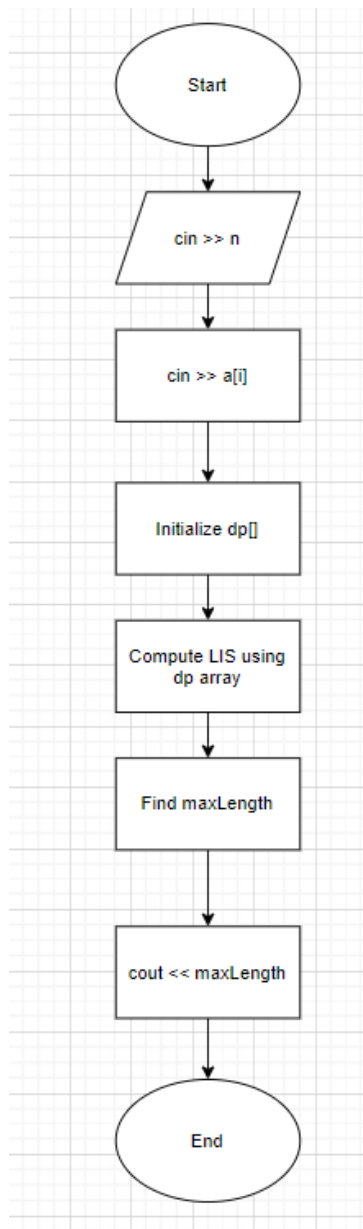


### Завдання №3 (Self Practice Work. Algotester “Найбільша зростаюча підпоследовність”)

Час виконання

Плановий: 1 година

Реальний: 45 хвилин

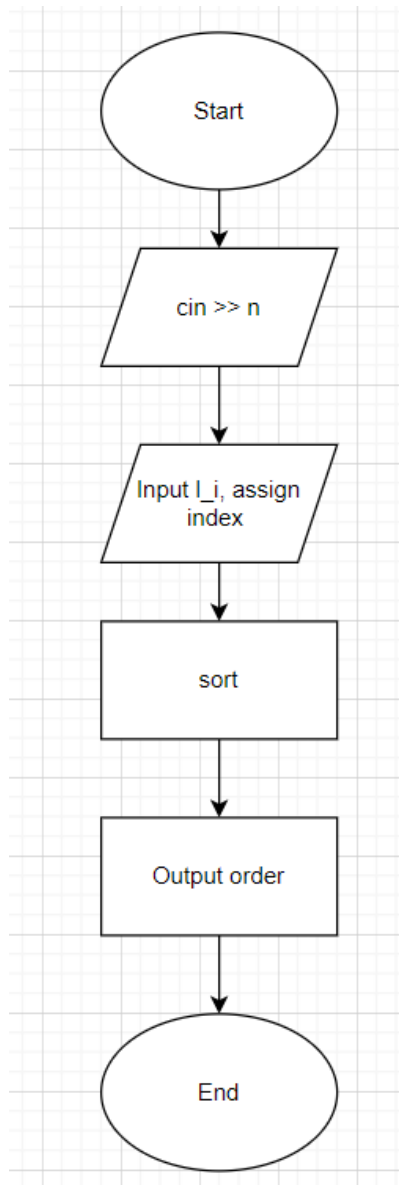


## Завдання №4 (Self Practice Work. Algotester “Офісна Вулиця. Частина 1”)

Час виконання

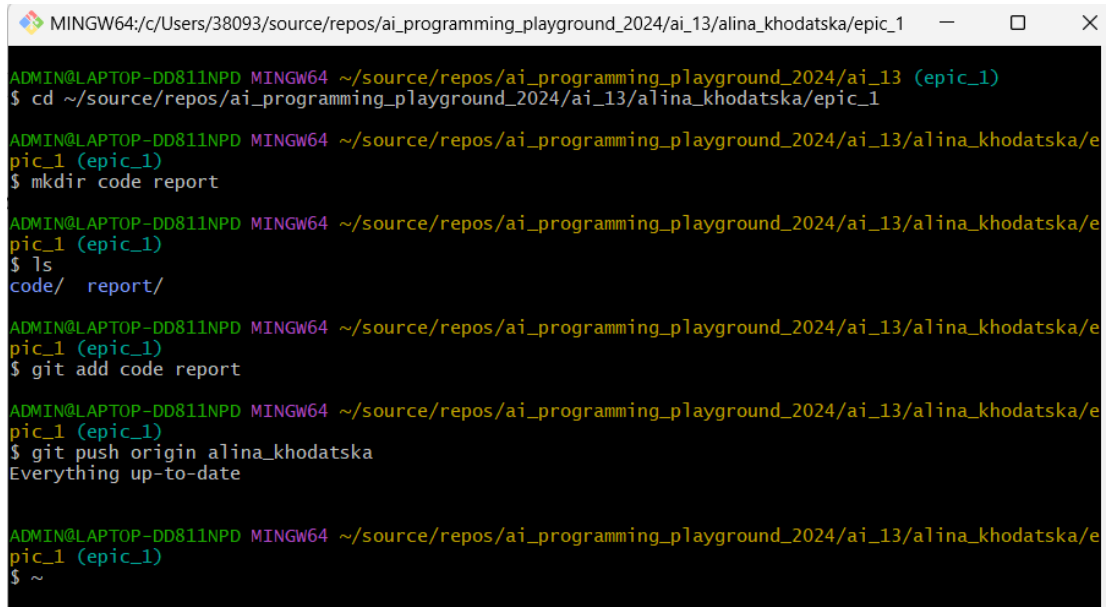
Плановий: 1 година

Реальний: 55 хвилин



### 3. Конфігурація середовища до виконання завдань:

#### *Персональна гілка в Git*



```
MINGW64:/c:/Users/38093/source/repos/ai_programming_playground_2024/ai_13/alina_khodatska/epic_1
ADMIN@LAPTOP-DD811NPD MINGW64 ~/source/repos/ai_programming_playground_2024/ai_13 (epic_1)
$ cd ~/source/repos/ai_programming_playground_2024/ai_13/alina_khodatska/epic_1

ADMIN@LAPTOP-DD811NPD MINGW64 ~/source/repos/ai_programming_playground_2024/ai_13/alina_khodatska/epic_1 (epic_1)
$ mkdir code report

ADMIN@LAPTOP-DD811NPD MINGW64 ~/source/repos/ai_programming_playground_2024/ai_13/alina_khodatska/epic_1 (epic_1)
$ ls
code/  report/

ADMIN@LAPTOP-DD811NPD MINGW64 ~/source/repos/ai_programming_playground_2024/ai_13/alina_khodatska/epic_1 (epic_1)
$ git add code report

ADMIN@LAPTOP-DD811NPD MINGW64 ~/source/repos/ai_programming_playground_2024/ai_13/alina_khodatska/epic_1 (epic_1)
$ git push origin alina_khodatska
Everything up-to-date

ADMIN@LAPTOP-DD811NPD MINGW64 ~/source/repos/ai_programming_playground_2024/ai_13/alina_khodatska/epic_1 (epic_1)
$ ~
```

Alina Khodatska - Epic 1

Status	COMPLETE
Assignees	A
Dates	Empty
Priority	High
Time Estimate	Empty
Track Time	Add time
Tags	Empty
Relationships	Empty

Custom Fields

+ Create Custom Field

Subtasks

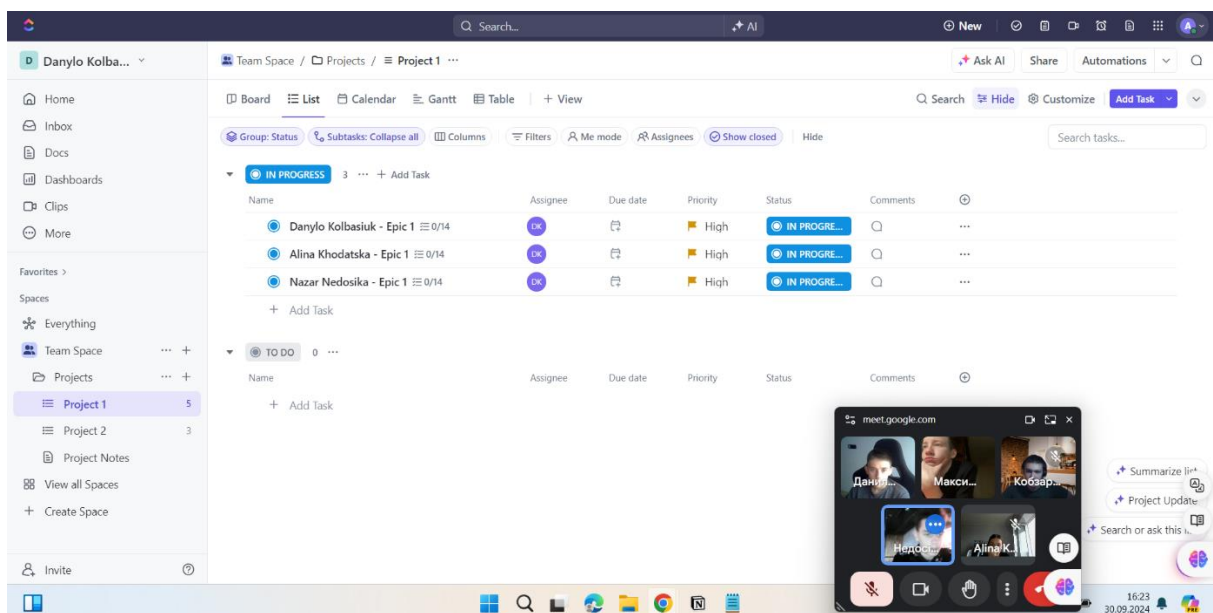
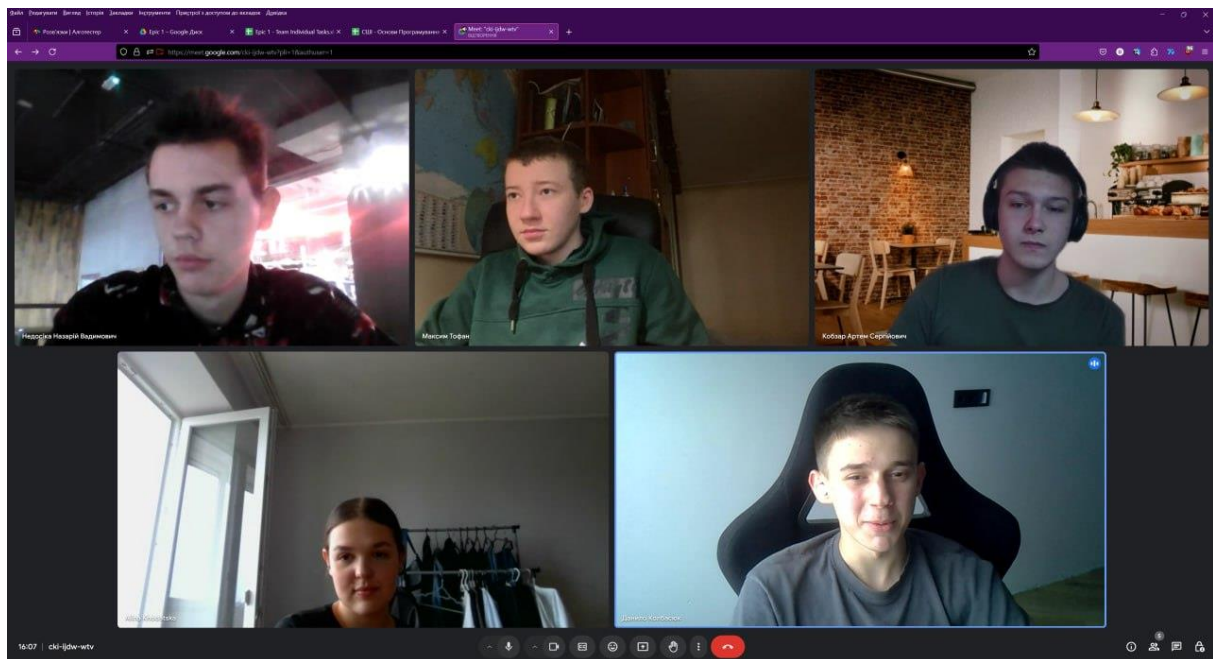
+ New Task

Checklists 14/14

Checklist (14/14)
✓ Package Managers-OS
✓ Console Commands
✓ VS-Code
✓ VS-Code Extensions
✓ Debugger & Linter

✓ Git
✓ GitHub
✓ GitHub PRs and Code Reviews
✓ Trello (or a different task manager)
✓ Algotester
✓ Flowcharts & Draw.io
✓ Word & Writing Reports
✓ Numeral systems & Practice with binary numbers
✓ First Program

## Зустріч з командою



**4. Код програм з посиланням на зовнішні ресурси:**

*До завдання №1:*

`practice_work_task_1_alina_khodatska.cpp`

*До завдання №2:*

`self_practice_work_algotester_task_1_alina_khodatska.cpp`

*До завдання №3:*

`self_practice_work_algotester_task_2_alina_khodatska.cpp`

*До завдання №4:*

`self_practice_work_algotester_task_3_alina_khodatska.cpp`

## 5. Результати виконання завдань та тестування

### Завдання №1 (Class Practice Work. Обчислення складних відсотків за депозитом)

```
#include <iostream>
#include <stdio.h>
#include <cmath>
using namespace std;
// (A) - загальна сума після інвестування
// (P) - початкова сума
// (r) - річна ставка
// (n) - кількість нарахувань відсотків за рік
// (t) - кількість років
// (ClearProfit) - чистий прибуток
int main()
{
    double A, P, r, ClearProfit;
    int n, t;
    printf("Enter the initial investment amount: ");
    if (scanf_s("%lf", &P) != 1) {
        printf("Invalid input for the initial investment amount.\n");
        return 1;
    }
    printf("Enter the annual interest rate (in decimal): ");
    if (scanf_s("%lf", &r) != 1) {
        printf("Invalid input for the annual interest rate.\n");
        return 1;
    }
    printf("Enter the number of times interest is compounded per year: ");
    if (scanf_s("%d", &n) != 1) {
        printf("Invalid input for the number of times interest is compounded.\n");
        return 1;
    }
    printf("Enter the number of years the money is invested: ");
    if (scanf_s("%d", &t) != 1) {
        printf("Invalid input for the number of years.\n");
        return 1;
    }
    A = P * pow(1 + r / n, n * t);
    ClearProfit = A - P;
    printf("Your total amount after investment: %lf\n", A);
    printf("Your initial investment amount: %lf\n", P);
    printf("Your clear profit amount is: %lf\n", ClearProfit);
    return 0;
}
```

```
Enter the initial investment amount: 5
Enter the annual interest rate (in decimal): 2
Enter the number of times interest is compounded per year: 12
Enter the number of years the money is invested: 10
Your total amount after investment: 540237541.089321
Your initial investment amount: 5.000000
Your clear profit amount is: 540237536.089321
```

## Завдання № 2 (Self Practice Work. Algotester “A+B”)

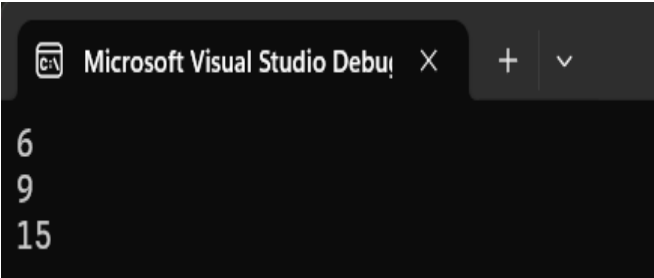
```
#include <iostream>
using namespace std;

int main() {
    int a, b;
    cin >> a >> b;

    int sum = a + b;

    cout << sum << endl;

    return 0;
}
```



Microsoft Visual Studio Debug Console output:

```
6
9
15
```

## Завдання №3 (Self Practice Work. Algotester “Найбільша зростаюча підпослідовність”)

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

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    vector<int> a(n);
    cout << "Enter the numbers: ";
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

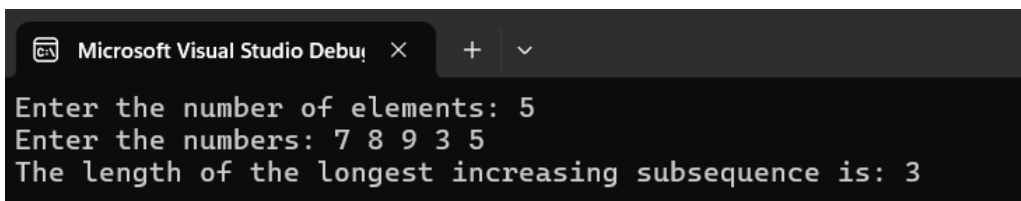
    vector<int> dp(n, 1);

    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (a[i] > a[j]) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
    }

    int maxLength = 0;
    for (int i = 0; i < n; i++) {
        maxLength = max(maxLength, dp[i]);
    }

    cout << "The length of the longest increasing subsequence is: " << maxLength << endl;

    return 0;
}
```



Microsoft Visual Studio Debug Console output:

```
Enter the number of elements: 5
Enter the numbers: 7 8 9 3 5
The length of the longest increasing subsequence is: 3
```



#### Завдання №4 (Self Practice Work. Algotester “ Офісна Вулиця. Частина 1”)

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main() {
    int n;
    cin >> n;

    vector<pair<int, int>> offices(n);

    for (int i = 0; i < n; i++) {
        cin >> offices[i].first;
        offices[i].second = i + 1;
    }

    sort(offices.begin(), offices.end());

    for (int i = 0; i < n; i++) {
        cout << offices[i].second << " ";
    }
    cout << endl;

    return 0;
}
```

Microsoft Visual Studio Debug Console

```
4 5 6 7 8
1 2 3 4
```

**Висновок:**

Після завершення цього епіку я змогла опанувати поняття Git і GitHub та їх використання, вивчила команди Linux, освоїла Visual Studio та компілятор. Я зрозуміла алгоритм обчислення в системах числення. Також виконувала додаткові завдання, щоб більше практикуватися з мовою C++.

**Посилання на pull request:**