

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



## **Звіт**

про виконання

**Лабораторних та практичних робіт №1**

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

**з розділу:** «Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні»

***Виконала:***

студент групи ШІ-13

Присліпська Софія Андріївна

Львів 2023

**Тема роботи:** Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні

### **Мета роботи:**

- 1) Теорія Навчальна діяльність.
- 2) Управління вимогами (розуміти завдання) та проектування (намалювати блок-схеми та оцінити завдання 3-7)
- 3) Лабораторна робота № програмування: VNS Lab 1 Завдання 1
- 4) Лабораторна робота № програмування: VNS Lab 1 Завдання 2
- 5) Лабораторна робота № програмування: Algotester Lab 1 Завдання 1
- 6) Практика # програмування: Практична задача класу
- 7) Практичне # програмування: завдання для самостійної практики
- 8) Звіт про документацію результатів і дії щодо розміщення результатів (документи та програми на GitHub)
- 9) Оцінка та публікація результатів

### **Теоретичні відомості:**

- 1) Теоретичні відомості з переліком важливих тем:
  - Тема №1: Умовні оператори, такі як if, else if і else
  - Тема №2: Константи та змінні
  - Тема №3: лінійні та розгалужені алгоритми
- 2) Індивідуальний план опрацювання теорії:
  - Тема №1: Умовні оператори, такі як if, else if і else
    - <https://youtu.be/Zkr0aAdD1Ww?feature=shared>
    - 
    - Статус: Ознайомлена
    - Початок опрацювання теми: 10.10
    - Звершення опрацювання теми: 10.10
    -
  - Тема №2: Константи та змінні
    - 
    - <https://youtu.be/mJzNnXia-mU?feature=shared>
    - 
    - Статус: Ознайомлена
    - Початок опрацювання теми: 26.09
    - Звершення опрацювання теми: 26.09
  - Тема №3: лінійні та розгалужені алгоритми
    - 
    - [https://youtu.be/EPfSVJkx\\_0?feature=shared](https://youtu.be/EPfSVJkx_0?feature=shared)

- Статус: Ознайомлена
- Початок опрацювання теми: 30.09
- Звершення опрацювання теми: 30.09

## Виконання роботи:

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

Завдання № 1 Lab# programming: VNS Lab 1 Task 1

- Варіант завдання:15
- Тема: "Знайомство з С. Виконання програми простої структури"
- Мета: Знайомство з середовищем програмування, створення, відлагодження й виконання простої програми, що містить ввід/вивід інформації й найпростіші обчислення.
- Деталі завдання:  
Обчислити значення виразу при різних дійсних типах даних (float й double).
- Обчислення варто виконувати з використанням проміжних змінних.
- Порівняти й пояснити отримані результати.
- 
- Важливі деталі для врахування в імплементації програми

Завдання №2 Lab# programming: VNS Lab 1 Task 2

- Варіант завдання:15
- Деталі завдання:
- Обчислити значення виразів. Пояснити отримані результати.

Завдання №3 Lab# programming: Algotester Lab 1 Task 1

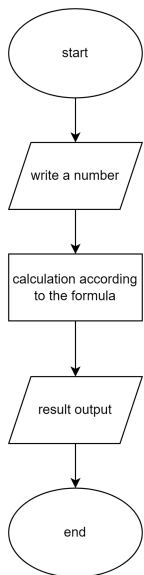
- Варіант завдання:2
- Деталі завдання:
- У вас є стіл, у якого є 4 ніжки, довжини яких вам дано.
- Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте  $d$  від кожної ніжки (тобто
- вам буде дано 4 числа, кожне з яких буде означати відпилювання від відповідної ніжки стола).
- Якщо під час відпилювання найдовша ніжка стола буде у 2 рази більша-рівна ніж найменша
- ніжка - стіл перевернеться, але відпилювати ніжки це вам не завадить.
- Тобто якщо  $h_{\max} \geq 2 * h_{\min}$  то стіл перевертається. Увага, це може статися і між початком
- та кінцем відпилювання, наприклад коли відпиляють 2, але ще не встигнуть відпиляти 3тю ніжку.
- Також ми вважаємо що перед відпилюванням стіл не перевернеться.
- Ваше завдання сказати чи після усіх маніпуляцій стіл буде цілий та паралельний підлозі.
- Якщо довжина, яку відріжуть буде більша за довжину ножки - вам треба вивести ERROR.
- Увага! Навіть якщо стіл перевернеться - ви все одно відпилюєте ніжки і можете отримати
- ERROR.
- 
- 
-

- 
- Завдання №4 Practice# programming: Class Practice Task
- Деталі завдання:
- 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!"

- Завдання №5 Practice# programming: Self Practice Task
- Деталі завдання:
- Програма виводить найменше та найбільше значення масиву, а також середнє арифметичне елементів

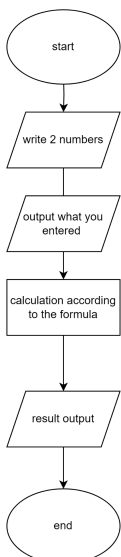
## 2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 Lab# programming: VNS Lab 1 Task 1



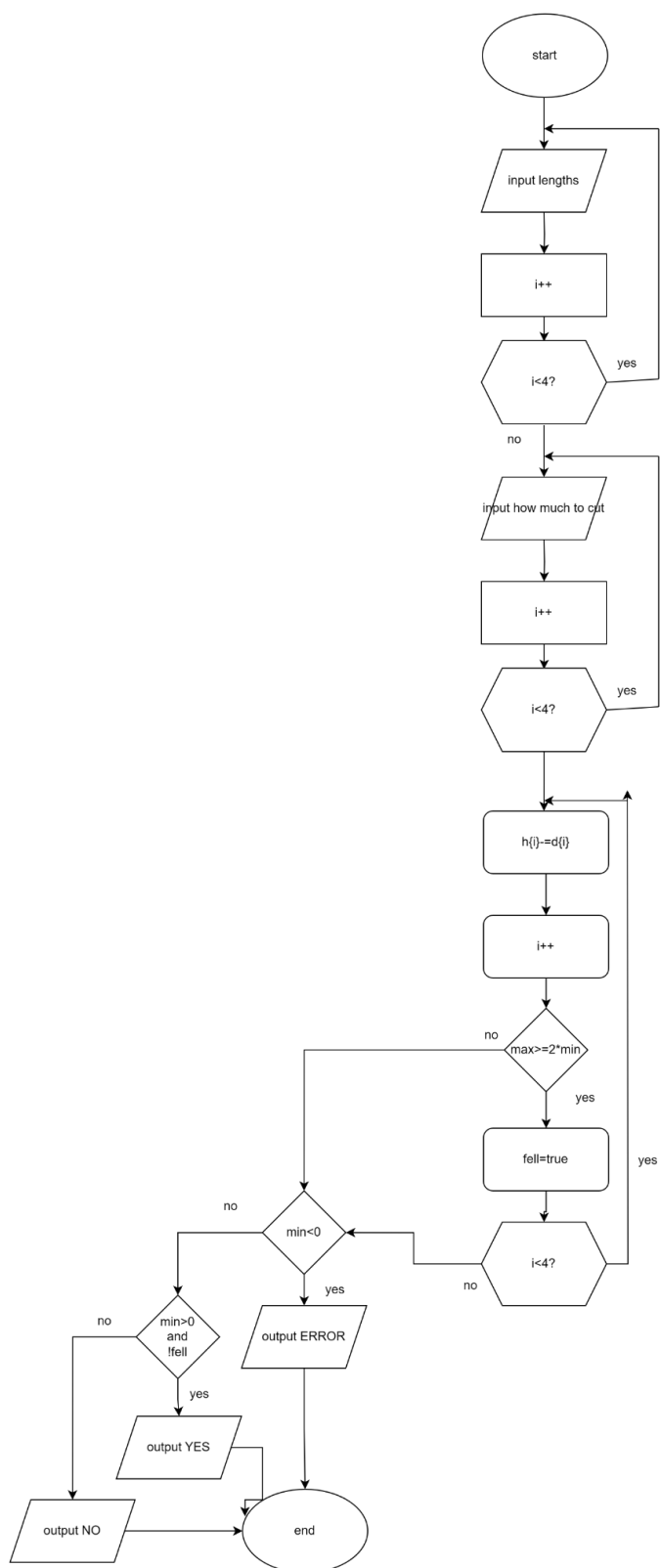
- рисунок 1: блок схема до програми №1
- Планований час на реалізацію: 1 година
- 

Програма №2 Lab# programming: VNS Lab 1 Task 2



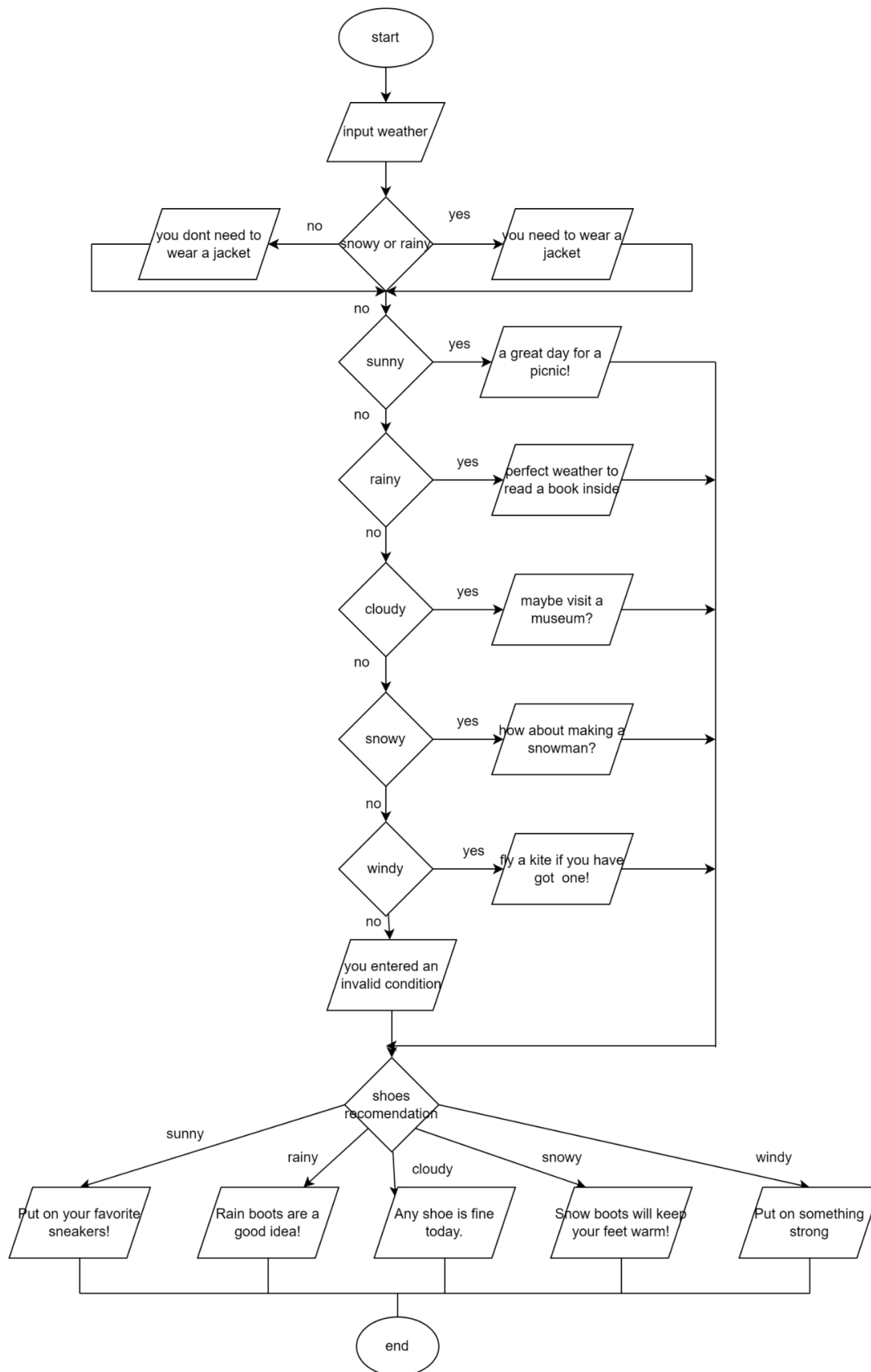
- рисунок 2: блок схема до програми №2
- Планований час на реалізацію: 1 година

# Програма №3 Lab# programming: Algotester Lab 1 Task 1



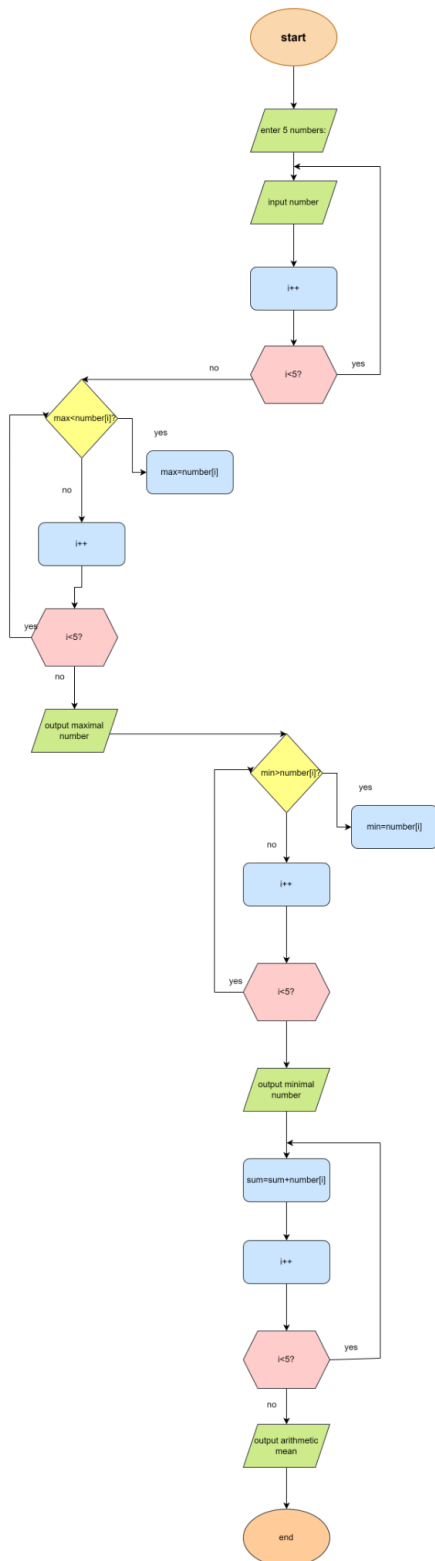
- рисунок 3: блок схема до програми №3
- Планований час на реалізацію:3 дні

# Програма №4 Practice# programming: Class Practice Task



- 
- рисунок 4: блок схема до програми №4
- Планований час на реалізацію:2 дні

## Програма №5 Practice# programming: Self Practice Task



- рисунок 5: блок схема до програми №5
- Планований час на реалізацію: 2 години



-

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

Завдання №1

```
#include <iostream>

#include <cmath>

using namespace std;

int main () {

double a=1000;

double b=0.0001;

double action1= pow(a+b, 3);

double action2=pow(a,3);

double action3=3*a*pow(b,2);

double action4=pow(b,3);

double action5=3*pow(a,2)*b;

double result=(action1-action2)/(action3+action4+action5);

cout <<result<< endl;


float a_f=1000;

float b_f=0.0001;

float action1_f= pow(a_f+b_f, 3);

float action2_f=pow(a_f,3);

float action3_f=3*a_f*pow(b_f,2);

float action4_f=pow(b_f,3);
```

```

float action5_f=3*pow(a_f,2)*b_f;

float result_f=(action1_f-action2_f)/(action3_f+action4_f+action5_f);

cout <<result_f<< endl;

return 0;

}

```

рисунок 1: код до програми №1

## Завдання №2

```

#include <iostream>

using namespace std;

int main()
{
    cout << "Enter a number: ";

    float m = 0;

    cin >> m;

    cout << "You entered " << m << endl;

    cout << "Enter a number: ";

    float n = 0;

    cin >> n;

    cout << "You entered " << n << endl;

    float result1=(m++-n);

    cout <<result1<<endl;
}

```

```

float result2=(m-->n);

cout <<result2<<endl;


float result3=(n-- >m);

cout <<result3<<endl;


return 0;

}

```

рисунок 2: код до програми №2

### Завдання №3

```

#include <bits/stdc++.h>

using namespace std;

long int h[4], d[4];

int main() {

    for(int i = 0; i < 4; ++i){

        cin >> h[i];

    }


    for (int i = 0; i < 4; ++i) {

```

```

        cin >> d[i];

    }

    long int n = sizeof(h) / sizeof(h[0]);

    bool fell = false;

    for (int i = 0; i < 4; ++i) {

        h[i] -= d[i];

        if (*max_element(h, h+n) >= 2 * *min_element(h, h+n)){

            fell = true;

        }

    }

    if (*min_element(h, h+n) < 0){

        cout << "ERROR" << endl;

    }

    else if(!fell && *min_element(h, h+n) > 0){

        cout << "YES" << endl;

    }

    else{

        cout << "NO" << endl;

    }

    return 0;

}

```

рисунок 3: код до програми №3

<https://www.geeksforgeeks.org/how-to-find-the-maximum-element-of-an-array-using-stl-in-c/>

#### Завдання №4

```
#include <iostream>

using namespace std;

#include <cmath>

int main()
{
    int weather;

    cout << "Enter the current weather conditions (1-sunny, 2-rainy, 3-cloudy, 4-snowy, 5-windy): " << endl;

    cin >> weather;

    if (weather == 2 || weather == 4) // snowy
    {
        cout << "you need to wear a jacket" << endl;
    }
    else
    {
        cout << "no need to wear a jacket" << endl;
    }

    if (weather == 1)
    {
        cout << "A great day for a picnic!" << endl;
    }

    else if (weather == 2)
```

```

{
    cout << "Perfect weather to read a book inside!" << endl;
}

else if (weather == 3)
{
    cout << "Maybe visit the museum?" << endl;
}

else if (weather == 4)
{
    cout << "How about making a snowman?" << endl;
}

else if (weather == 5)
{
    cout << "Fly the kite if you have one!" << endl;
}

else
{
    cout << "you entered an invalid condition!" << endl
        << "please try again" << endl;

    return 0;
}

switch (weather)
{
case 1:
    cout << "Put on your favorite sneakers!" << endl;

```

```

        break;

    case 2:

        cout << "Rain boots are a good idea!" << endl;

        break;

    case 3:

        cout << "Any shoe is fine today." << endl;

        break;

    case 4:

        cout << "Snow boots will keep your feet warm!" << endl;

        break;

    case 5:

        cout << "Put on something strong!" << endl;

        break;

    }

    return 0;
}

```

рисунок 4: код до програми №4

#### Завдання №5

```

#include <iostream>

using namespace std;

int main()
{
    int amount = 5;

```

```
int number[5];

cout << "enter 5 numbers:" << endl;

for (int i = 0; i < 5; i++)
{
    cin >> number[i];
}

int max = number[0];

for (int i = 0; i < 5; i++)
{
    if (max < number[i])
    {
        max = number[i];
    }
}

cout << "the maximal number is:" << max << endl;

int min = number[0];

for (int i = 0; i < 5; i++)
{
    if (min > number[i])
    {
        min = number[i];
    }
}

cout << "the minimal number is:" << min << endl;
```



```
double sum=0;

for (int i = 0; i <5; i++)

{

    sum=sum+number[i];

}

cout<<sum<<endl;

cout<<"the arithmetic mean of your numbers:"<<sum/5;

return 0;

}
```

рисунок 5: код до програми №5

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

Завдання №1

output:

1

1.28

Час затрачених на виконання завдання:1 година

Завдання №2

output:

Enter a number: 5

You entered 5

Enter a number: 8

You entered 8

-3

0

1

Час затрачений на виконання завдання:1 година

Завдання №3

<https://algotester.com/en/ProblemSolution/Display/1503777>

Час затрачений на виконання завдання:1 тиждень

Завдання №4

output:

Enter the current weather conditions (1-sunny, 2-rainy, 3-cloudy, 4-snowy, 5-windy):

2

you need to wear a jacket

Perfect weather to read a book inside!

Rain boots are a good idea!

Час затрачений на виконання завдання:1 день

Завдання №5

output:

enter 5 numbers:

9

7

10

8

15

the maximal number is:15

the minimal number is:7

49the arithmetic mean of your numbers:9.8

Час затрачений на виконання завдання:2 години

## **Висновки:**

В ході виконання лабораторних та практичних робіт з теми "Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні" я реалізувала на практиці свої теоретичні знання про умовні оператори, константи, змінні, лінійні та розгалужені алгоритми. Створила програму для отримання рекомендацій щодо активностей на основі введених користувачем погодних умов, також дві програми що обчислюють певні вирази, ще одна: для пошуку максимальних та мінімальних чисел серед введених та їхнє середнє арифметичне число. Створила програму для визначення стану стола після відпилювання частини ніжок. Програма перевіряє, чи буде стіл цілим та паралельним підлозі після маніпуляцій. До кожної з цих задач я створила відповідну блок схему.