

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



Звіт

про виконання лабораторних та практичних робіт блоку № 2

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 № 2

Виконав:

Студент групи ШІ-13
Литвин Маркіян Назарович

Львів 2024

Тема: Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід. Базові операції та вбудовані функції. Коментарі.

Мета: навчитися ефективно будувати програмну логіку та працювати з різними типами даних, створювати чіткі й структуровані програми, що вирішують різноманітні задачі з використанням стандартних інструментів програмування.

Теоретичні відомості:

1. Системи числення:

- Визначення та види систем числення (двійкова, вісімкова, десяткова, шістнадцяткова).
- Перетворення чисел між системами.
- Практичне застосування систем числення в програмуванні.
- Вправи на перетворення чисел між системами.

2. Компіляція:

- Етапи компіляції.
- Препроцесор і директива `include`.
- Функції компілятора.

3. Змінні, Константи, Типи даних та їх Розміри:

- Визначення та оголошення змінних і констант.
- Основні типи даних у C++ (`int`, `char`, `float`, `double`, тощо).
- Розміри типів даних і їх представлення у двійковій системі.
- Порівняння різних типів даних та вибір відповідного типу для задач.

4. Бібліотеки в C++

- Поняття бібліотеки;
- Використання бібліотеки;
- Бібліотеки C++ - `stdio`, `cmath`, `iostream` (Стандартна бібліотека C++)

5. Ввід та Вивід даних:

- Основи використання `cin` та `cout`.
- Основи використання `printf/scanf`.
- Форматування виводу даних.
- Робота з різними типами даних у вводі/виводі.
- Практичні вправи на ввід та вивід даних.

6. Базові Операції та Вбудовані Функції:

- Арифметичні операції та їх використання.
- Побітові операції
- Використання математичних функцій (sqrt, pow, тощо).
- Вправи на роботу з базовими операціями та функціями.

7. Коментарі у Коді:

- Важливість коментарів у програмуванні.
- Види коментарів у C++.
- Написання ефективних коментарів.
- Практика коментування коду.

8. Лінійні алгоритми:

- Визначення та особливості лінійних алгоритмів.
- Структура та властивості лінійних алгоритмів.
- Написання лінійних алгоритмів на C++.
- Вправи на створення лінійних алгоритмів.

9. Розгалужені алгоритми та Умовні Оператори:

- Введення в розгалужені алгоритми.
- Область видимості.
- Умовні оператори: if, else, else if, switch-case.
- Тернарний оператор ?.
- Реалізація розгалужених алгоритмів на C++.
- Практичні задачі на розгалужені алгоритми.

10. Логічні Оператори:

- Визначення та види логічних операторів (AND, OR, NOT).
- Застосування логічних операторів у розгалужених алгоритмах.
- Практичні приклади використання логічних операторів.
- Вправи на розуміння та використання логічних операцій.

Індивідуальний план опрацювання теорії:

Системи числення

Компілятор

Змінні та типи даних

Бібліотеки C++

Ввід/вивід

Вбудовані функції

Коментарі

Лінійні алгоритми
Розгалужені алгоритми та Умовні Оператори
Логічні Оператори

Джерела:

- <https://www.youtube.com/watch?v=KymlNuMfcV4>
- <https://acode.com.ua/uroki-po-cpp/#toc-0>
- **Chatgpt**
- **Лекційний матеріал**

Виконання роботи:

Завдання №1: VNS Lab 1

Варіант 9

1. Обчислити значення виразу при різних дійсних типах даних (float й double).

Обчислення варто виконувати з використанням проміжних змінних.

Порівняти й пояснити отримані результати.

2. Обчислити значення виразів. Пояснити отримані результати.

Task 1

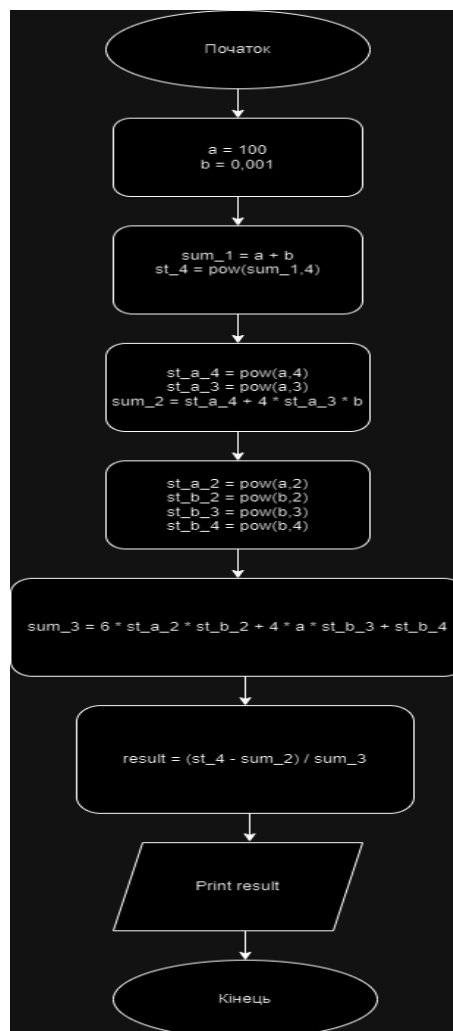
9	$\frac{(a+b)^4 - (a^4 + 4a^3b)}{6a^2b^2 + 4ab^3 + b^4},$ при a=100, b=0.001
---	--

Task 2

- 1) ++n*++m
- 2) m++<n
- 3) n++>m

Task 1:

Блок-схема:



Розв'язок:

```
1  #include <iostream>
2  #include <math.h>
3  using namespace std;
4
5  int main(){
6  int choice;
7  cout << "Введіть float - 1 чи double - 2: ";
8  cin >> choice;
9  if (choice == 1){
10     float a1 = 100;
11     float b1 = 0.001;
12     float sum1_1 = a1+b1;
13     float st1_4 = pow(sum1_1,4);
14     float st_a1_4 = pow(a1,4);
15     float st_a1_3 = pow(a1,3);
16     float st_a1_2 = pow(a1,2);
17     float st_b1_2 = pow(b1,2);
18     float st_b1_3 = pow(b1,3);
19     float st_b1_4 = pow(b1,4);
20     float sum1_2 = (st_a1_4 + 4 * st_a1_3 * b1);
21     float sum1_3 = (6 * st_a1_2 * st_b1_2 + 4 * a1 * st_b1_3 + st_b1_4);
22     float result1 = (st1_4 - sum1_2) / sum1_3;
23     cout << "Float result: " << result1 << endl;
24 } else if (choice==2){
25     double a2 = 100;
26     double b2 = 0.001;
27     double sum2_1 = a2+b2;
28     double st2_4 = pow(sum2_1,4);
29     double st_a2_4 = pow(a2,4);
30     double st_a2_3 = pow(a2,3);
31     double st_a2_2 = pow(a2,2);
32     double st_b2_2 = pow(b2,2);
33     double st_b2_3 = pow(b2,3);
34     double st_b2_4 = pow(b2,4);
35     double sum2_2 = (st_a2_4 + 4 * st_a2_3 * b2);
36     double sum2_3 = (6 * st_a2_2 * st_b2_2 + 4 * a2 * st_b2_3 + st_b2_4);
37     double result2 = (st2_4 - sum2_2) / sum2_3;
38     cout << "Double result: " << result2 << endl;
39     return 0;
40 }
41
```

Результат:

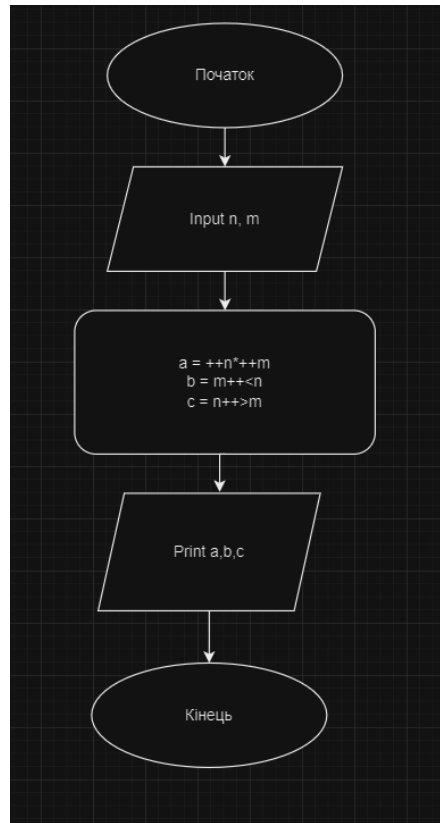
```
Введіть float - 1 чи double - 2: 1
Float result: 0
```

```
Введіть float - 1 чи double - 2: 2
Double result: 1
```

Час виконання 15 хв

Task 2:

Блок-схема:



Розв'язок:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      long long int n,m;
7      cout << "Введіть n: ";
8      cin >> n;
9      cout << "Введіть m: ";
10     cin >> m;
11     long long a = ++n * ++m;
12     bool b = m++ < n;
13     bool c = n++ > m;
14     cout << "a: " << a << " " << "b: " << b << " " << "c: " << c << endl;
15     cout << "n: " << n << " " << "m: " << m; }
```

Результат:

```
Введіть n: 12
Введіть m: 10
a: 143 b: 1 c: 1
n: 14 m: 12
```

Час виконання 10 хв.

Завдання №2: Algotester Lab 1:

Варіант - 1

У вас є стіл, у якого є 4 ніжки, довжини яких вам дано.

Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте dd 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.

Input

4 цілих числа $h_{1,2,3,4}$ - довжини ніжок стола 4 цілих числа $d_{1,2,3,4}$ - довжина, яку відпиляють від відповідної ножки

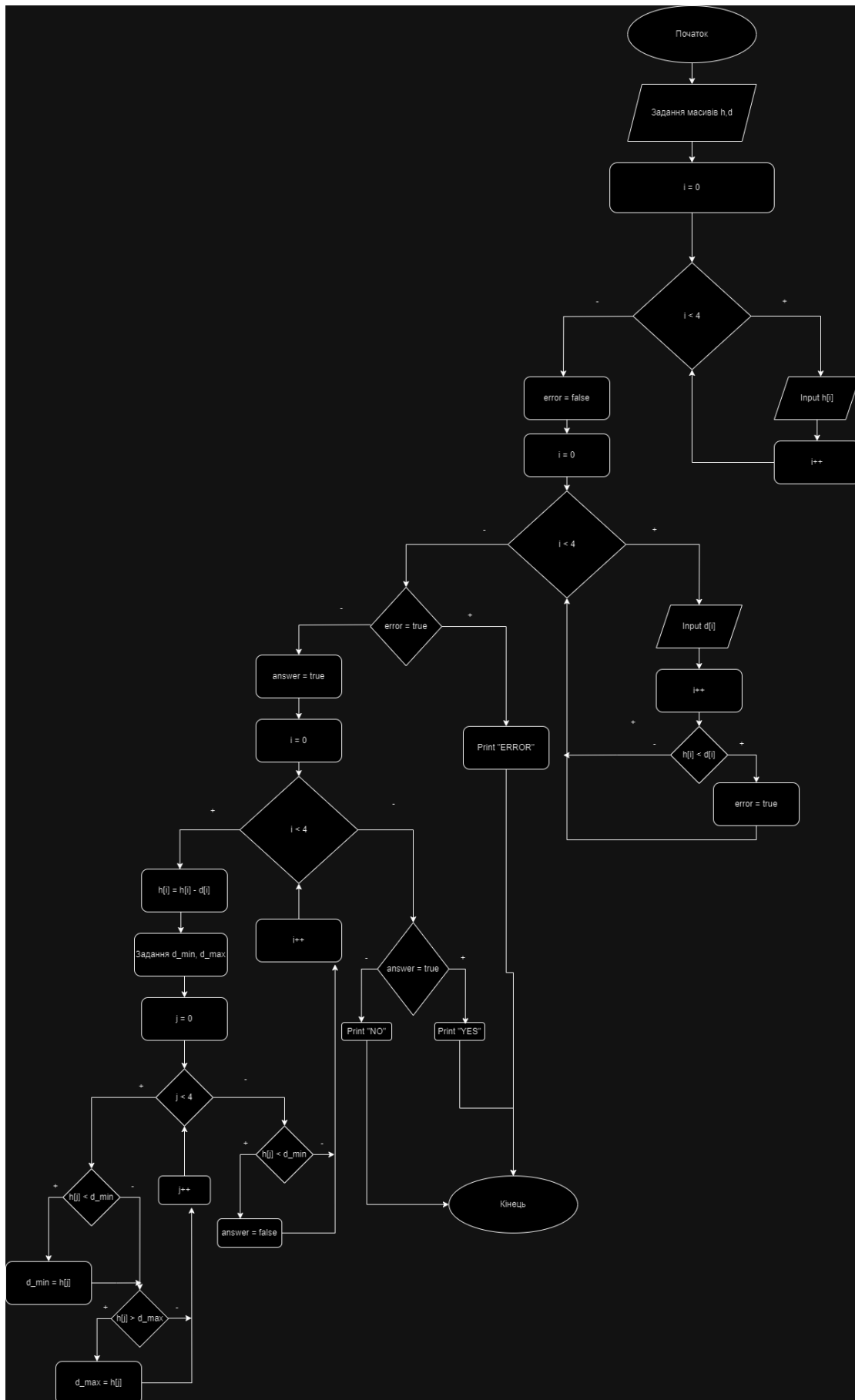
Output

YES - якщо стіл буде стояти паралельно площині підлоги та довжина найменшої ніжки не буде рівна нулю.

ERROR - у випадку якщо ви відпиляете більшу довжину ніж має ножка

NO - у інших випадках

Блок-схема:



Розв'язок:

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      long long h[4], d[4];
6      for (int i = 0; i < 4; i++){
7          cin >> h[i];
8      }
9      bool error = false;
10     for (int i = 0; i < 4; i++){
11         cin >> d[i];
12         if (h[i] < d[i])
13             error = true;
14     }
15     if (!error){
16         bool answer = true;
17         for (int i = 0; i < 4; i++){
18             h[i] = h[i] - d[i];
19             long long d_max = h[0], d_min = h[0];
20             for (int j = 1; j < 4; j++){
21                 if (h[j] < d_min){
22                     d_min = h[j];
23                 }
24                 if (h[j] > d_max){
25                     d_max = h[j];
26                 }
27                 if (d_max >= 2 * d_min) {
28                     answer = false;
29                     break;
30                 }
31             }
32         }
33         if (answer){
34             cout << "YES";
35         }
36         else {
37             cout << "NO";
38         }
39     }
40     else {
41         cout << "ERROR";
42     }
43     return 0;
44 }
```

Результат:

```
10
10
10
10
5
1
1
1
NO
```

2 hours ago

C++ 23

Accepted

0.003

1,422 View

Час виконання: 1.5 години

Завдання №3: Practice Task:

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

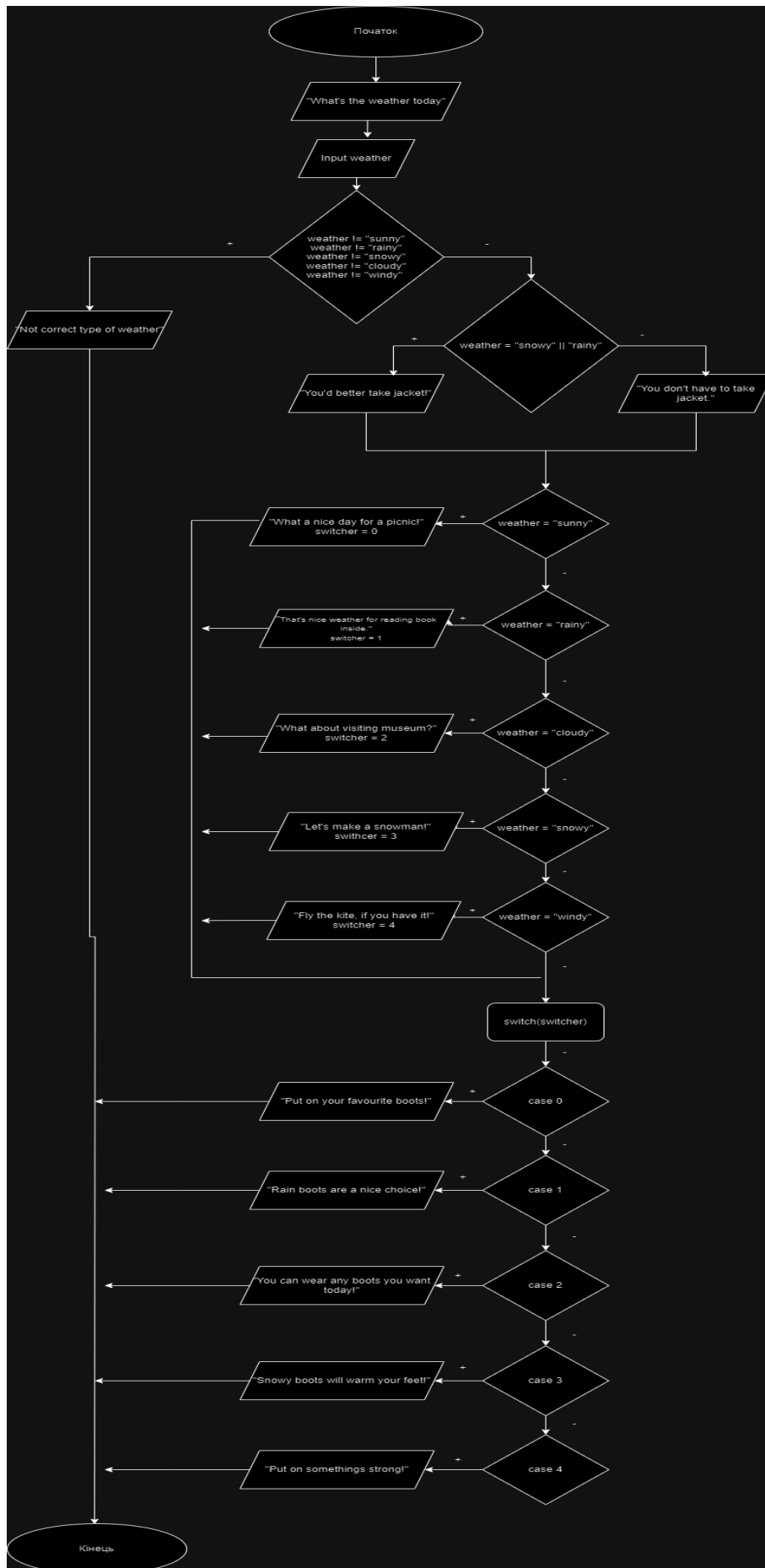
Можливі варіанти погоди:

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

Вимоги:

1. Використати всі згадані в передумові задачі оператори галуження - *if else*, *if, else if*, *switch case*;
2. За потреби комбінувати оператори;

Блок-схема:



Розв'язок:

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main(){
6     int switcher;
7     string weather;
8     cout << "What's the weather today? (sunny, rainy, cloudy, snowy, windy)" << endl;
9     cin >> weather;
10    if (weather != "sunny" && weather != "rainy" && weather != "cloudy" && weather != "snowy" && weather != "windy"){
11        cout << "Not correct type of weather." << endl;
12    }
13    if (weather == "sunny" || weather == "rainy" || weather == "cloudy" || weather == "snowy" || weather == "windy"){
14        if (weather == "rainy" || weather == "snowy") {
15            cout << "You'd better take jacket!" << endl;
16        }
17        else{
18            cout << "You don't have to take jacket." << endl;
19        }
20        if (weather == "sunny"){
21            cout << "What a nice day for a picnic!" << endl;
22            switcher=0;
23        }
24        else if(weather == "rainy"){
25            cout << "That's nice weather for reading book inside." << endl;
26            switcher=1;
27        }
28        else if (weather == "cloudy"){
29            cout << "What about visiting museum?" << endl;
30            switcher=2;
31        }
32        else if (weather == "snowy"){
33            cout << "Let's make a snowman!" << endl;
34            switcher=3;
35        }
36        else if (weather == "windy"){
37            cout << "Fly the kite, if you have it!" << endl;
38            switcher=4;
39        }
40    }
41    switch(switcher){
42        case 0:
43            if (weather == "sunny"){
44                cout << "Put on your favourite boots!";
45                break;
46            }
47        case 1:
48            if (weather == "rainy"){
49                cout << "Rain boots are a nice choice!";
50                break;
51            }
52        case 2:
53            if (weather == "cloudy"){
54                cout << "You can wear any boots you want today!";
55                break;
56            }
57        case 3:
58            if (weather == "snowy"){
59                cout << "Snowy boots will warm your feet!";
60                break;
61            }
62        case 4:
63            if (weather == "windy"){
64                cout << "Put on somethings strong!";
65                break;
66            }
67        default:
68            break;
69    }
70    return 0;
71 }
```

Результат:

```
What's the weather today? (sunny, rainy, cloudy, snowy, windy)
rainy
You'd better take jacket!
That's nice weather for reading book inside.
Rain boots are a nice choice!
```

Час виконання: 35 хв

Завдання №4: Self Algotester Task:

У вашого персонажа є H хітпойнтів та M мани.

Персонаж 3 рази використає закляття, кожне з яких може використати хітпойнти та ману одночасно.

Якщо якийсь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні **АБО** хітпойнти, **АБО** ману.

Якщо в кінці персонаж буде мати додатню кількість хітпойнтів та мани ($H, M > 0$) - він виграє, в іншому випадку програє.

Ваше завдання у випадку виграшу персонажа вивести **YES**, вивести **NO** у іншому випадку.

Input

2 цілих числа HH та MM - хітпойнти та мана персонажа

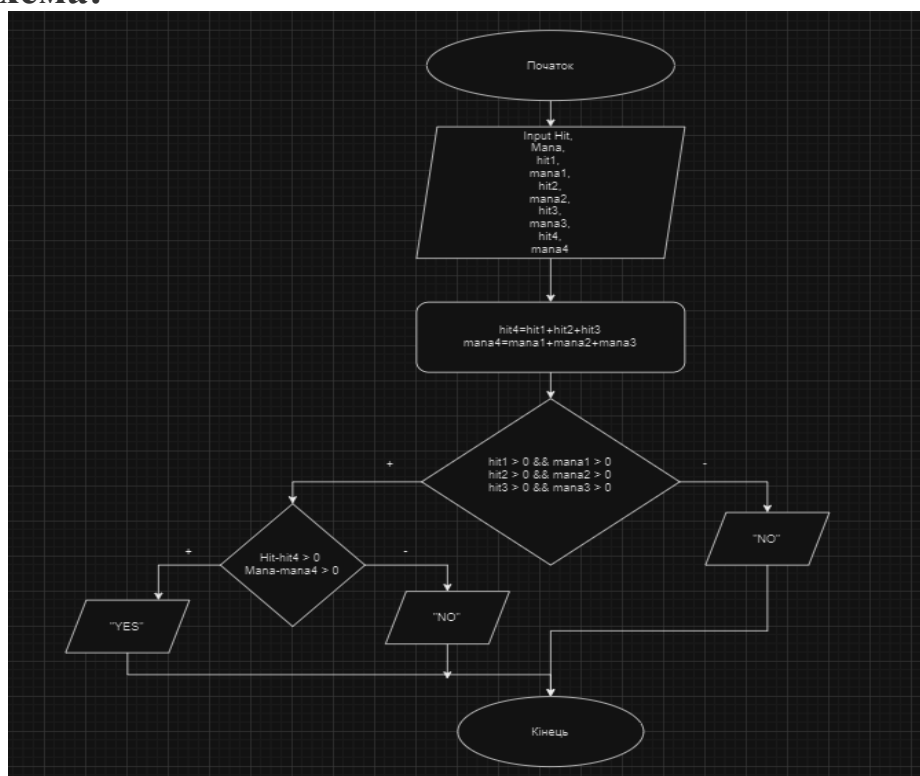
3 рядки по 2 цілих числа, hi та mi - кількість хітпойнтів та мани, які ваш персонаж потратить за хід на i заклинання

Output

YESYES - якщо ваш персонаж виграє

NONO - у всіх інших випадках

Блок-схема:



Розв'язок:

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      long long Hit, Mana, hit1, mana1, hit2, mana2, hit3, mana3, hit4, mana4;
6      cin >> Hit >> Mana;
7      cin >> hit1 >> mana1;
8      cin >> hit2 >> mana2;
9      cin >> hit3 >> mana3;
10     hit4=hit1+hit2+hit3;
11     mana4=mana1+mana2+mana3;
12     if ((hit1 > 0 && mana1 > 0) || (hit2 > 0 && mana2 > 0) || (hit3 > 0 && mana3 > 0)){
13         cout << "NO";}
14     else if (Hit-hit4 > 0 && Mana-mana4 > 0){
15         cout << "YES";}
16     else{
17         cout << "NO";}}
```

Результат:

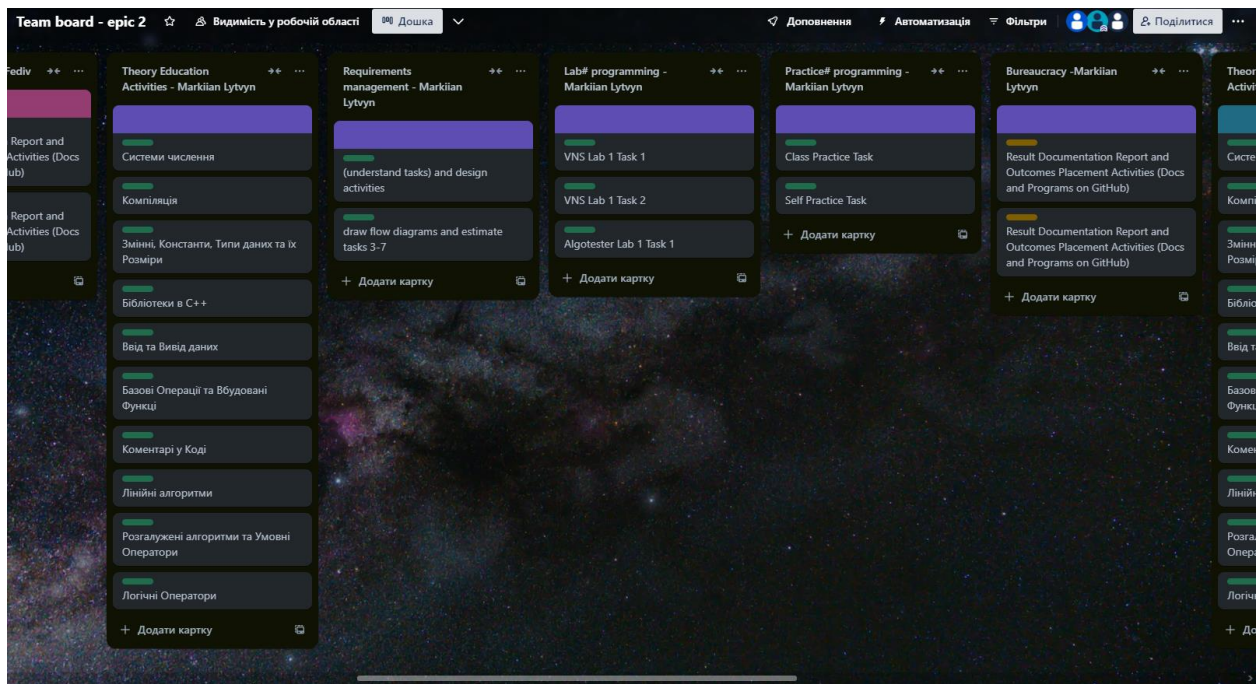
```
100
100
10
0
10
0
79
0
YES
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
26 minutes ago	C++ 23	Accepted	0.003	1.203	View

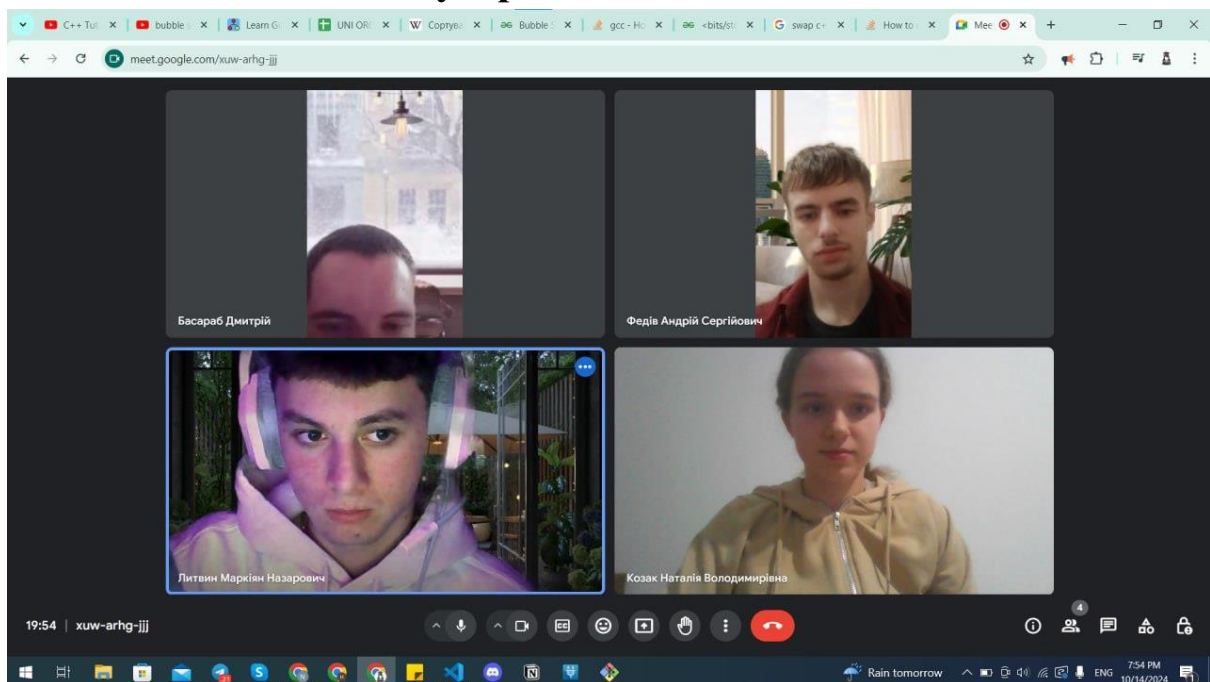
Час виконання: 25 хв

Team work:

Trello



Зустрічі з командою



Висновок:

Виконуючи цей епік я навчився користуватись операторами if-else та switch, поглибив свої знання з систем числення, компілятора, змінних та типів даних, бібліотеки C++, вводу/виводу, вбудованих функцій, лінійних алгоритмів, розгалужених алгоритмів та операторів.