Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №1 та №2**

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

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

***Виконала:***

студентка групи ШІ-13

Кузнєцова Анастасія Володимирівна

# **Тема роботи:**

Побудова лінійних та розгалужених алгоритмів. Застосування умовних операторів. Освоєння та використання змінних, констант, операцій вводу та виводу. Ознайомлення з масивами.

# **Мета роботи:**

Вміти будувати лінійні та розгалужені алгоритми, застосовувати умовні оператори if, else if, else, while та switch case. Освоїти змінні й константи.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Лінійні та розгалужені алгоритми.
* Тема №2: Оператори if, else if, else.
* Тема №5: Оператор switch.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Лінійні та розгалужені алгоритми.
  + Джерела Інформації:
    - Стаття: https://disted.edu.vn.ua/courses/learn/10348
  + Що опрацьовано:
    - Опрацьовано види алгоритмів та їх структуру.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.10.23
  + Звершення опрацювання теми: 18.10.23
* Тема №2: Оператори if, else if, else, while.
  + Джерела Інформації:
    - Відео:   
      <https://youtu.be/zogwWqGyM2c?si=v93nzb4vR8ASkISH>  
      https://youtu.be/ckJtOMcIxyU?si=PI2nHmhSnG17iZNT
    - Стаття: https://acode.com.ua/urok-67-operatory-umovnogo-rozgaluzhennya-if-else/
  + Що опрацьовано:
    - Опрацьовано інформацію про оператори if, else if, else і яким чином їх використовувати.
    - Ознайомлено з оператором while та як його застосовувати.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.10.23
  + Звершення опрацювання теми: 30.10.23
* Тема №3: Оператор switch.
  + Джерела Інформації:
    - Відео: https://www.youtube.com/watch?v=1BnNEWA9RWI
    - Стаття: https://acode.com.ua/urok-68-operator-switch/
  + Що опрацьовано:
    - Опрацьовано інформацію про оператор switch case та його особливості.
    - Використання оператора в практичній роботі.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.10.23
  + Звершення опрацювання теми: 02.11.23

# **Виконання роботи:**

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

Завдання №1 VNS Lab №1

* Варіант завдання: 18

Task 1:  
- Деталі завдання: порахувати і вивести значення за формулою при різних типах даних.  
- Важливі деталі для врахування в імплементації програми: використати типи даних float і double та пояснити різницю у відповідях. Виконувати обчислення, використовуючи проміжні змінні.

Task 2:   
- Деталі завдання: обчислити значення виразів та пояснити отримані результати.

Завдання №2 Algotester Lab 1

* Варіант завдання: 1
* Деталі завдання: у першому рядку 2 цiлих числа H та M - хiтпойнти та мана персонажа. 3 рядки по 2 цiлих числа, hi та mi - кiлькiсть хiтпойнтiв та мани, якi персонаж витратить за хiд на i-те заклинання. Вивести YES, якщо персонаж виграє, NO – у всіх інших випадках.
* Важливі деталі для врахування в імплементації програми: звернути увагу на обмеження 1≤H, M, hi, mi≤.

Завдання №3 Practice work

* Деталі завдання: Створення простого пораднику щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди.
* Важливі деталі для врахування в імплементації програми: Запропоновані погодні умови: sunny, rainy, cloudy, snowy, windy. Якщо користувач вводить будь-яку іншу умову, йому пропонується ввести дійсну умову.

Завдання №4 Self-practice №1

* Варіант завдання: «Куди бігти?»
* Деталі завдання: У першому рядку задано три цілих числа через пробіл sd, su та v — відстань до першої та другої локацій, і звичайна швидкість геймерів. У єдиному рядку вивести куди варто рухатися хлопцям: вгору (Up) або вниз (Down), щоб найшвидше дійти до локації. Якщо час, за який вони можуть дійти до обох локацій рівний, виведсти Never mind.
* Важливі деталі для врахування в імплементації програми: звернути увагу на обмеження 0≤sd,su,v≤

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

Програма №1 VNS Lab №1 Task 1

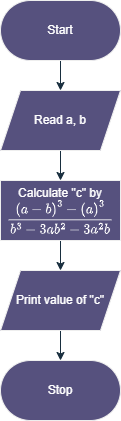


Рисунок 1: Блок схема до програми №1

* Планований час на реалізацію: 1 годину
* Важливі деталі для врахування в імплементації: використати типи даних float і double та пояснити різницю у відповідях. Виконувати обчислення, використовуючи проміжні змінні.

Програма №2 VNS Lab №1 Task 2

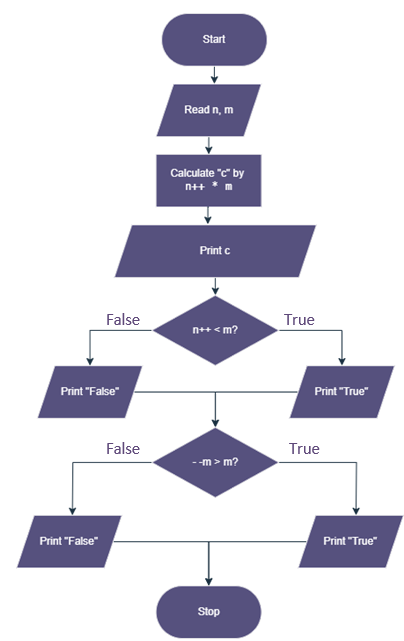


Рисунок 2: Блок схема до програми №2

* Планований час на реалізацію: 1 годину

Програма №3 Algotester Lab 1

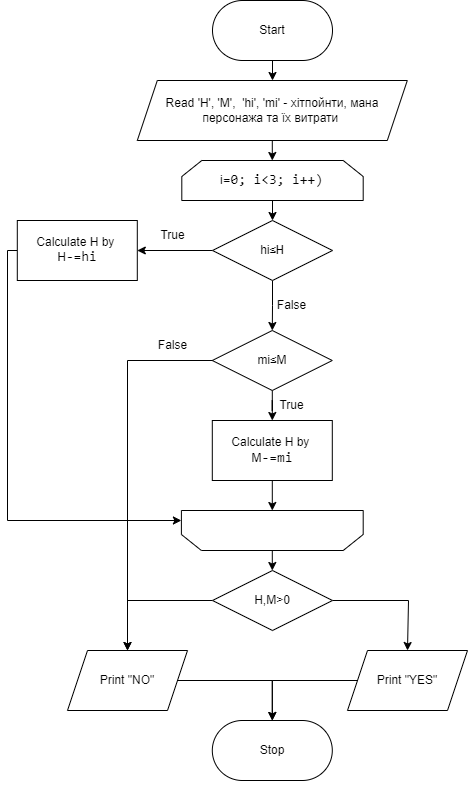


Рисунок 3: Блок схема до програми №3

* Планований час на реалізацію: 4 години
* Важливі деталі для врахування в імплементації: звернути увагу на обмеження 1≤H, M, hi, mi≤

Програма №4 Practice work

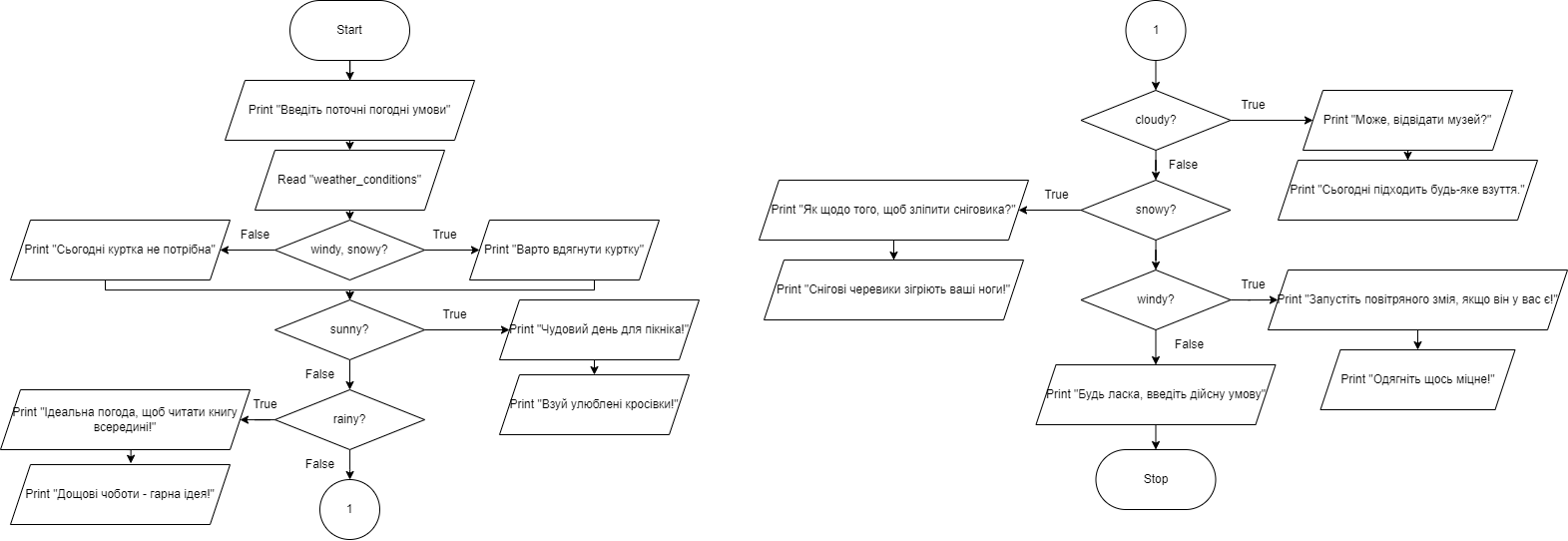


Рисунок 4: Блок схема до програми №4

* Планований час на реалізацію: 3 години
* Важливі деталі для врахування в імплементації: Запропоновані погодні умови: sunny, rainy, cloudy, snowy, windy. Якщо користувач вводить будь-яку іншу умову, йому пропонується ввести дійсну умову.

Програма №5 Self-practice №1

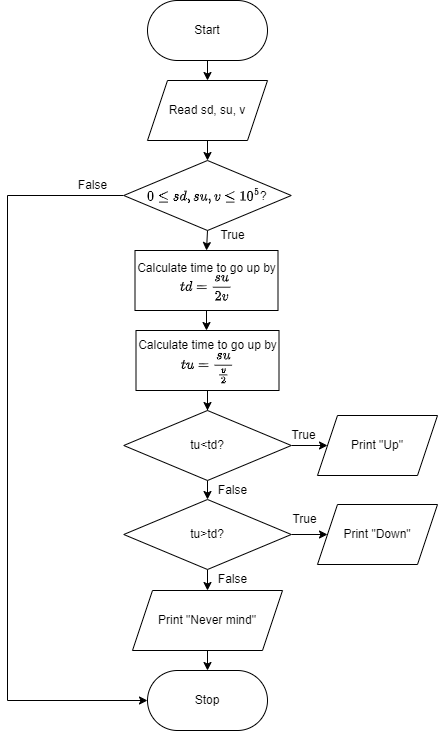


Рисунок 5: Блок схема до програми №5

* Планований час на реалізацію: 4 години
* Важливі деталі для врахування в імплементації: звернути увагу на обмеження 0≤sd,su,v≤

## **3. Конфігурація середовища до виконання завдань:**

Файл settings.json в якому зберігаються налаштування VS Code

{

*"workbench.colorTheme"*: "Monokai",

*"code-runner.runInTerminal"*: true,

*"C\_Cpp.default.compilerPath"*: "d:\\C C++\\test.c",

*"cmake.configureOnOpen"*: true,

*"terminal.integrated.defaultProfile.windows"*: "Windows PowerShell",

*"terminal.integrated.profiles.windows"*: {

*"PowerShell"*: {

*"source"*: "PowerShell",

*"icon"*: "terminal-powershell"

        },

*"Command Prompt"*: {

*"path"*: [

                "${env:windir}\\Sysnative\\cmd.exe",

                "${env:windir}\\System32\\cmd.exe"

            ],

*"args"*: [],

*"icon"*: "terminal-cmd"

        },

*"Git Bash"*: {

*"source"*: "Git Bash"

        },

*"bash (MSYS2)"*: {

*"path"*: "C:\\msys64\\usr\\bin\\bash.exe",

*"args"*: [

                "--login",

                "-i"

            ]

        },

*"Windows PowerShell"*: {

*"path"*: "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"

        }

    },

*"workbench.colorCustomizations"*: {}

}

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

Завдання №1 VNS Lab №1

Task 1

#include <iostream>

#include <math.h>

using *namespace* std;

*int* main ()

{

*double* a = 1000.0;

*double* b = 0.0001;

*double* c;

*double* d = pow((a-b), 3), e = pow(a, 3), f = pow(b, 3), g = pow(b, 2), h = pow(a, 2);

    c = (d-e) / (f-3\*a\*g -3\*h\*b);

//c = (pow((a-b), 3)-(pow(a, 3))) / (pow(b, 3)-3\*a\*pow(b, 2) -3\*pow(a, 2)\*b);

    cout <<c;

    return 0;

}

VNS Lab 1 Task 1 : using double

Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/208/files#diff-4f9ca9080fedc49aac20c930ed3dcab02e8fb8c145395c3389e25340d0ba5094>

#include <iostream>

#include <math.h>

using *namespace* std;

*int* main ()

{

*float* a = 1000.0;

*float* b = 0.0001;

*float* c;

*float* d = pow((a-b), 3), e = pow(a, 3), f = pow(b, 3), g = pow(b, 2), h = pow(a, 2);

    c = (d-e) / (f-3\*a\*g -3\*h\*b);

//c = (pow((a-b), 3)-(pow(a, 3))) / (pow(b, 3)-3\*a\*pow(b, 2) -3\*pow(a, 2)\*b);

    cout<<c;

    return 0;

}

VNS Lab 1 Task 1 : using double

Завдання полягало в тому, щоб написати код, який зможе порахувати і вивести значення за заданою у варіанті формулою. Обов’язково треба перевірити відповідь з різними типами даних: float та double. Для обрахунку за формулою використовую додаткові змінні, для зручності, при написанні формули.

Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/208/files#diff-40032c210db0ee57e11633f7e440c83fe012547405b19d28d50858476daa7fe4>

Task 2

#include <iostream>

using *namespace* std;

*int* main ()

{

*int* n, m, c;

    cout<<"Enter a first number\n";

    cin>>n;

    cout<<"Enter a second number\n";

    cin>>m;

   c = n++ \* m ;

    cout << "The answer is " << c << '\n';

cout << "For n++ < m the eqation is ";

    if (n++ < m)

        {

        cout<<"True\n";

        }

    else

        {

        cout<< "False\n";

        }

cout << "For --m > m the eqation is ";

     if (--m > m)

        {

        cout<<"True\n";

        }

    else

        {

        cout<< "False\n";

        }

return 0;

}

Завдання полягало в тому, щоб обрахувати та вивести чи правильне твердження наведене у варіанті з використанням користувацьких чисел.

Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/208/files#diff-40032c210db0ee57e11633f7e440c83fe012547405b19d28d50858476daa7fe4>

Завдання №2 Algotester Lab 1

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main()

{

*long* *long* H, M, hi, mi;

    cin>>H;

    cin>>M;

    if (H>pow(10, 12)||H<1||M>pow(10, 12)||M<1)

    {

        return 0;

    }

    for (*int* i=0; i<3; i++)

    {

        cin>>hi;

        cin>>mi;

        if(hi>pow(10, 12)||hi<0||mi>pow(10, 12)||mi<0)

        {

        return 0;

        }

        else if (hi>0 && mi>0)

        {

            break;

        }

        else

        {

            H-=hi;

            M-=mi;

        }

    }

    if (hi>0 && mi>0)

    {

        cout<<"NO";

    }

    else if (H>0 && M>0)

    {

        cout << "YES\n";

    }

    else

    {

        cout << "NO\n";

    }

return 0;

}

Спочатку запрошую ввід 2 цiлих чисел H та M - хiтпойнтів та мани персонажа. Далі користувач вводить 3 рядки по 2 цiлих числа, hi та mi - кiлькiсть хiтпойнтiв та мани, якi його персонаж витратить за хiд на i-те заклинання. Для цього я використала цикл. Якщо ж гравець вводить дані, які не входять у допустимі значення програма виводить «Введені данні не входять у обмеження» та завершує роботу. Якщо гравець використовує і ману і хітпойнти – цикл завершується і код видає “NO”. Якщо використано або ману або хітпойнти – вираховується кількість того, що вийде після їх використання. Якщо в кінці хітпойнти та мана лишились програма видає “YES”, що означає перемогу. Якщо ж значення дорівнює мінусовому числу, або 0, то код видає “NO”.

Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/208/files#diff-4fa23da1f576c0bae1c66127bb2c85bb57506bf0c21f6aa8de4a8f0d8225791d>

Завдання №3 Practice work

#include <iostream>

#include <string>

using *namespace* std;

*int* main()

{

string weather\_conditions;

cout<<"Введіть поточні погодні умови\n";

cin>>weather\_conditions;

if (weather\_conditions == "windy" || weather\_conditions == "snowy")

    {

    cout<<"Варто вдягнути куртку\n";

    }

else

    {

    cout<<"Сьогодні куртка не потрібна\n";

    }

if (weather\_conditions == "sunny")

{

    cout<<"Чудовий день для пікніка!\n";

}

else if (weather\_conditions == "rainy")

{

    cout<<"Ідеальна погода, щоб читати книгу всередині!\n";

}

else if (weather\_conditions == "cloudy")

{

    cout<<"Може, відвідати музей?\n";

}

else if (weather\_conditions == "snowy")

{

    cout<<"Як щодо того, щоб зліпити сніговика?\n";

}

else if (weather\_conditions == "windy")

{

    cout<<"Запустіть повітряного змія, якщо він y вас є!\n";

}

else

{

 cout<<"Будь ласка, введіть дійсну умову";

}

switch (weather\_conditions[0])

{

    case 's':

        if (weather\_conditions[1] == 'u')

        cout<<"Взуй улюблені кросівки!";

        else if (weather\_conditions[1] == 'n')

        cout<<"Снігові черевики зігріють ваші ноги!";

        break;

    case 'r':

        cout<<"Дощові чоботи - гарна ідея!";

        break;

    case 'c':

        cout<<"Сьогодні підходить будь-яке взуття.";

        break;

    case 'w':

        cout<<"Одягніть щось міцне!";

        break;

    default:

        cout<<"Будь ласка, введіть дійсну умову";

}

return 0;

}

Завдання полягало у тому, щоб створити простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди. Для цього я використала оператори умови. У switch case я використала масив, так як у цьому операторі не можна використовувати тип даний string, тому я створила масив з перших букв слів. Для слів sunny та snowy я використала додаткову умову про другі літери слова, щоб оператор міг розрізнити ці дві слова.

Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/208/files#diff-1a6898d0ef18a71e856731ace6ca5a9781c853e04f98d197b553e986c5df5c6e>

Завдання №4 Self-practice №1

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main()

{

*double* sd, su, v, td, tu;

    cin>>sd>>su>>v; //Відстань до першої та другої локацій, звичайна швидківсть геймерів

    if (sd<0 || sd>pow(10, 5) || su<0 || su>pow(10, 5) || v<0 || v>pow(10, 5))

    {

        return 0;

    }

    td = sd / (2 \* v);

    tu = su / (v / 2);

    if(tu<td)

    {

        cout<<"Up";

    }

    else if(td<tu)

    {

        cout<<"Down";

    }

    else

    {

        cout<<"Never mind";

    }

return 0;

}

Завдання полягало у тому, щоб вивести куди варто рухатися хлопцям: вгору (Up) або вниз (Down), щоб найшвидше дійти до локації; якщо час, за який вони можуть дійти до обох локацій рівний, виведсти Never mind. У першому рядку вводяться відстань вгору, вниз та швидкість. Якщо введені дані перевищують обмеження, програма завершує роботу. Далі за допомогою формули код вираховує час вгору та вниз, враховуючи, що при підйомі вгору швидкість зменшується 2 рази, а вниз збільшується в 2 рази. Якщо час вгору більший за вниз програма виводить «Down», якщо навпаки «Up», а якщо час рівний, то «Never mind».

Pull request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/208/files#diff-6f5c693309f857d2423a170491879e3ef29fd263651c60601e25bab8f749ab9d>

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

Завдання №1VNS Lab №1

Task 1

Коли використовуємо тип float

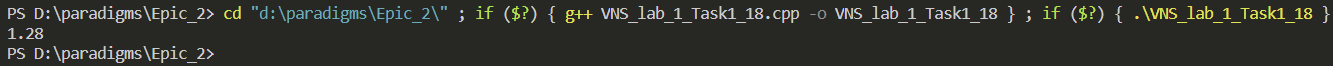


Рисунок 6 Термінал до програми №1: використання float

Коли використовуємо тип double



Рисунок 7 Термінал до програми №1: використання double

Різниця у відповіді через те, що тип float не достатньо точний на відміну від double для розрахунку. Тип float використовує 32 біти, а double 64, чому він і передає більшу кількість знаків після коми чітко і без округлення. Тому коли нам треба більш точні обчислення, треба передавати перевагу double.

Час, затрачений на виконання завдання: 1,5 години

Task 2

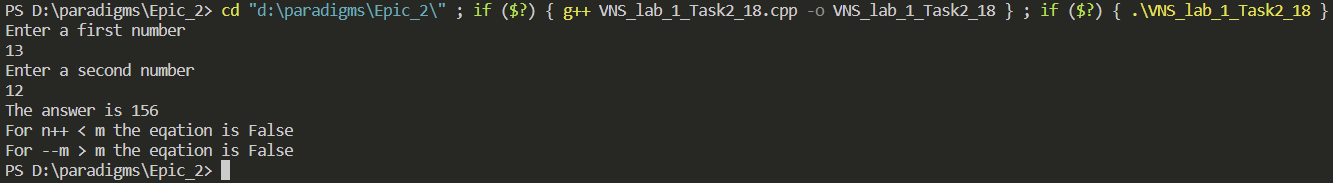


Рисунок 8 Термінал до програми №2

У вихідних даних: перший рядок видає добуток, порахований за формулою c = n++ \* m, коли n=13, а m=12. Відповідь така, бо спочатку n множиться на m, а потім вже до n додається 1 і воно стає 14.   
Другий рядок: відповідь false, бо спочатку n порівнюється з m (13<12), а потім до n додається один, по “++” це постфікс.  
Третій рядок: відповідь false, бо спочатку від m віднімається 1, бо “- -“ це префікс (m=11), а потім вже відбувається порівняння (11>12).

Час, затрачений на виконання завдання: 1 година

Завдання №2 Algotester Lab 1

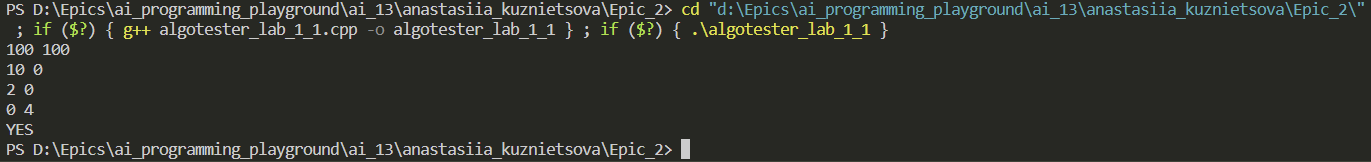


Рисунок 9 Термінал до програми №3

Перший рядок вхідних даних є кількість хітпойнтів та мани персонажа; другий – три пари чисел – витрати хітпойнтів та мани.  
Коди видає YES, бо після виконаних операцій у персонажа лишились мана та хітпойнти.

Час, затрачений на виконання завдання: 2 дні

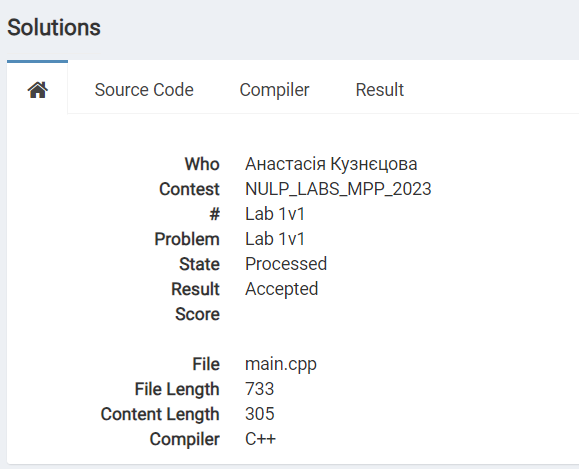


Рисунок 10 Скріншот з алготестера для Algotester lab 1

Завдання №3 Practice work

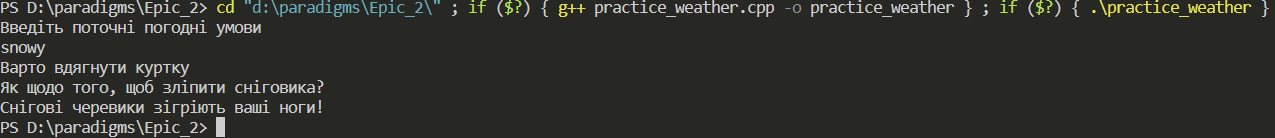


Рисунок 11 Термінал до програми №4

Після введення snowy код видає, за вказаними умовами, що варто вдягнути куртку та те, що варто зліпити сніговика. Останнім рядком є саме відповідь, про зимові черевики, бо наступною після “s” йде “n”.  
Час, затрачений на виконання завдання: 6 годин

Завдання №4 Self-practice

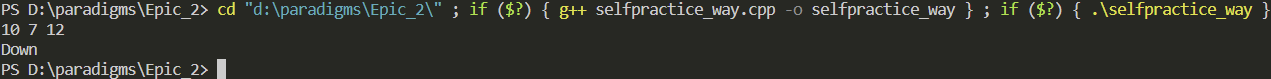


Рисунок 12 Термінал до програми №5

Вхідними даними є 10 – відстань до локації вниз, 7 – відстань до локації вгору та 12 – звичайна швидкість гравців.   
Відповідь “Down”, тому, що при порахунку часу, який вони витратять вгору чи вниз, час вниз вийшов менше, чому, внаслідок вказаних умов, було виведено, що варто йти вниз. Також було враховано, що коли гравці піднімаються вгору їх швидкість зменшується вдвічі, а вниз – збільшується вдвічі.

Час, затрачений на виконання завдання: 5 годин

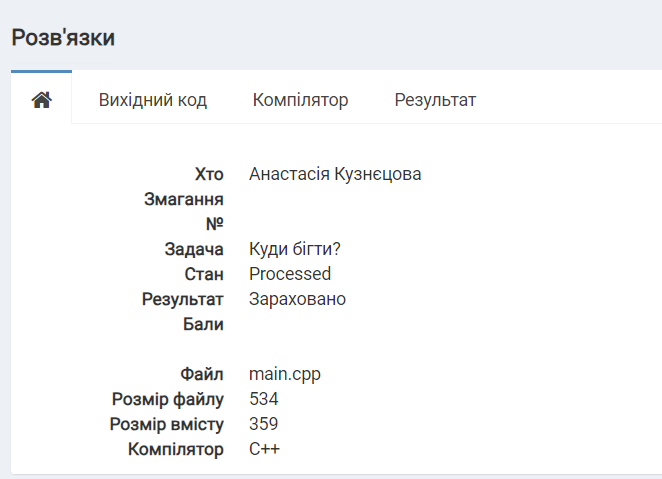


Рисунок 13 Скріншот з алготестера для завдання Self Practice

# **Висновки:**

Отже, я освоїла змінні й константи. Також я навчилась будувати лінійні та розгалужені алгоритми, застосовувати умовні оператори if, else if, else, while та switch case. Навчилась створювати масиви та використовувати вектори.