Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №(VNS LAB 1(Task 1,2),**

**Algotester LAB 1(Task 1)**

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

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

***Виконала:***

студентка групи ШІ-12

Бричко Марія Ігорівна

# **Тема роботи:**

Написання лабораторних робіт на мові С++ у VS Code. Створення блок-схем до коду. Застосування теоретичних знань з епіку 1 на практиці. Глибше вивчення мови С++: константи, змінні, лінійні та розгалужені алгоритми, умовні оператори, цикли

# **Мета роботи:**

Написати лабораторні роботи на мові С++ у VS Code. Створити блок-схеми до коду. Застосувати теоретичні знання з епіку 1 на практиці. Глибше вивчити мову С++: константи, змінні, лінійні та розгалужені алгоритми, умовні оператори, цикли

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Умовні оператори
* Тема №2: Константи, змінні
* Тема №3: Цикли
* Тема №4: Лінійні та розгалужені алгоритми

1. Індивідуальний план опрацювання теорії:

* Тема №1: Умовні оператори
  + Джерела Інформації
    - Стаття 1: <https://www.w3schools.com/cpp/cpp_conditions.asp>
    - Стаття 2: <https://acode.com.ua/urok-67-operatory-umovnogo-rozgaluzhennya-if-else/>
    - Стаття 3: <https://www.w3schools.com/cpp/cpp_switch.asp>
  + Що опрацьовано:
    - Умовні розгалуження if/else
    - Використання декількох операцій в розгалуженнях if/else
    - Неявне вказування блоків
    - Поєднання стейтментів if
    - Вкладені розгалуження if/else
    - Використання логічних операторів в розгалуженнях if/else
    - Основні використання розгалужень if/else
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.10.2023
  + Звершення опрацювання теми: 13.10.2023
* Тема №2: Константи, змінні
  + Джерела Інформації:
    - Стаття 1: <https://t.me/c/1980904348/1/299>
    - Стаття 2: <https://www.w3schools.com/cpp/cpp_variables.asp>
  + Що опрацьовано:
    - Оголошення декількох змінних, ідентифікатори, константи
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.10.2023
  + Звершення опрацювання теми: 14.10.2023
* Тема №3: Цикли
  + Джерела Інформації:
    - Стаття 1: <https://www.w3schools.com/cpp/cpp_while_loop.asp>
    - Стаття 2: <https://acode.com.ua/urok-71-tsykl-do-while/>
    - Стаття 3: <https://www.w3schools.com/cpp/cpp_for_loop.asp>
  + Що опрацьовано:
    - Цикл while, цикл do while, цикл for, оператори break і continue
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.10.2023
  + Звершення опрацювання теми: 18.10.2023
* Тема №4 Лінійні та розгалужені алгоритми
  + Джерела Інформації:
    - Стаття: <https://dystosvita.org.ua/course/view.php?id=13>
  + Що опрацьовано:
    - Основні поняття алгоритмізації
    - Лінійні алгоритми
    - Алгоритми з розгалуженнями
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 19.10.2023
  + Звершення опрацювання теми: 21.10.2023

# **Виконання роботи:**

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

Завдання №1 ”Practice\_Epic\_2”

* Варіант: Загальний варіант
* Деталі завдання: Створити простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди. Запропонувати користувачеві ввести поточні погодні умови: sunny; rainy; cloudy; snowy; windy. Якщо користувач вводить будь-яку іншу умову, запропонувати йому ввести дійсну умову. Якщо йде сніг або дощ, користувач повинен одягнути куртку. В іншому випадку куртка не потрібна. Якщо сонячно, порекомендувати «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, порекомендувати «Як щодо того, щоб зліпити сніговика?». Інакше, якщо буде вітер, порекомендувати «Запустіть повітряного змія, якщо він у вас є!». Рекомендації щодо взуття: 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!"
* Важливі деталі для врахування в імплементації програми: Перевірити значення погоди (воно має бути одним з запропонованих варіантів: sunny; rainy; cloudy; snowy; windy.

Завдання №2 “VNS\_LAB\_1\_1”

* Варіант завдання: 17
* Деталі завдання: Обчислити значення виразу за формулою при різних дійсних типах даних (float й double). Обчислення виконати з використанням проміжних змінних. Вивести результат.
* Важливі деталі для врахування в імплементації програми: Перевірити чи всі змінні у першому випадку мають тип даних float, і у другому - double.

Завдання №3 “VNS\_LAB\_1\_2”

* Варіант завдання: 17
* Деталі завдання: Обчислити значення заданих виразів
* Важливі деталі для врахування в імплементації програми: Важливо чи ми маємо префіксний чи постфіксний інкремент/декремент.

Завдання №4 “VNS\_LAB\_2”

* Варіант завдання: 9
* Деталі завдання: Використовуючи оператор циклу, знайти суму ряду з точністю ε=0.0001, за даною формулою. Вивести результат.
* Важливі деталі для врахування в імплементації програми: При визначенні суми членів ряду потрібно знайти рекурентну формулу для отримання наступного члена ряду.

Завдання №5 “Algotester\_lab1”

* Варіант завдання: 1
* Деталі завдання: У персонажа є H хiтпойнтiв та M мани. Персонаж 3 рази використає закляття, кожне з яких може використати хiтпойнти та ману одночасно. Якщо якесь закляття забирає i хiтпойнти i ману - персонаж програє. Якщо в кiнцi персонаж буде мати додатню кiлькiсть хiтпойнтiв та мани – вiн виграє, в iншому випадку програє. У випадку виграшу персонажа вивести YES, вивести NO у iншому випадку.
* Важливі деталі для врахування в імплементації програми: Для виграшу треба використовувати при одному заклинаннi АБО хiтпойнти, АБО ману. Також необхідно врахувати наступні обмеження:

1 ≤ H ≤ 1012

1 ≤ M ≤ 1012

0 ≤ hi ≤ 1012

0 ≤ mi ≤ 1012

Завдання №6 “Algotester\_lab2”

* Варіант завдання: 1
* Деталі завдання: Є дорога, яка виглядає як N чисел. Пiсля того як по нiй проїхати - втому можна визначити як рiзницю максимального та мiнiмального елементу. Потрібно мiнiмiзувати втому, але все що можна зробити - викинути одне число з дороги,

тобто забрати його з масиву. В результатi цiєї дiї, обчислити яку мiнiмальну втому можна отримати в кiнцi дороги.

* Важливі деталі для врахування в імплементації програми: Необхідно врахувати наступні обмеження:

1 ≤ N ≤ 105

0 ≤ ri ≤ 105

Завдання №7 “Self\_Practice\_Task”

* Варіант завдання: без варіанту
* Деталі завдання: В колодi знаходиться 2 · n карт. На кожнiй картi написано число вiд 1 до 2 · n. Числа на всiх картах рiзнi. В гру грають двоє гравцiв. На початку гри гравцi перемiшують карти в колодi i кожен з гравцiв отримує n карт. Гра складається з n раундiв. В кожному раундi гравцi обирають по однiй зi своїх ще не зiграних карт i одночасно кладуть на стiл. Раунд виграє той гравець, який поставив карту з бiльшим числом. Переможець гри — той, хто виграв бiльшу кiлькiсть раундiв. Гравцi уже перемiшали колоду i роздали карти. Ви добре знаєте свого суперника i його стратегiю — послiдовнiсть карт, якими вiн ходитиме. Визнчити чи є у вас можливiсть перемогти у грi. Вивести слово Possible, якщо ви можете перемогти, або Impossible, якщо не можете.
* Важливі деталі для врахування в імплементації програми: Необхідно врахувати наступні обмеження:
* 1 ≤ n ≤ 105,
* n — непарне,
* об’єднання масивiв a i b формує перестановку чисел вiд 1 до 2 · n.

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

Програма №1 ”Practice\_Epic\_2”

* Блок-схема

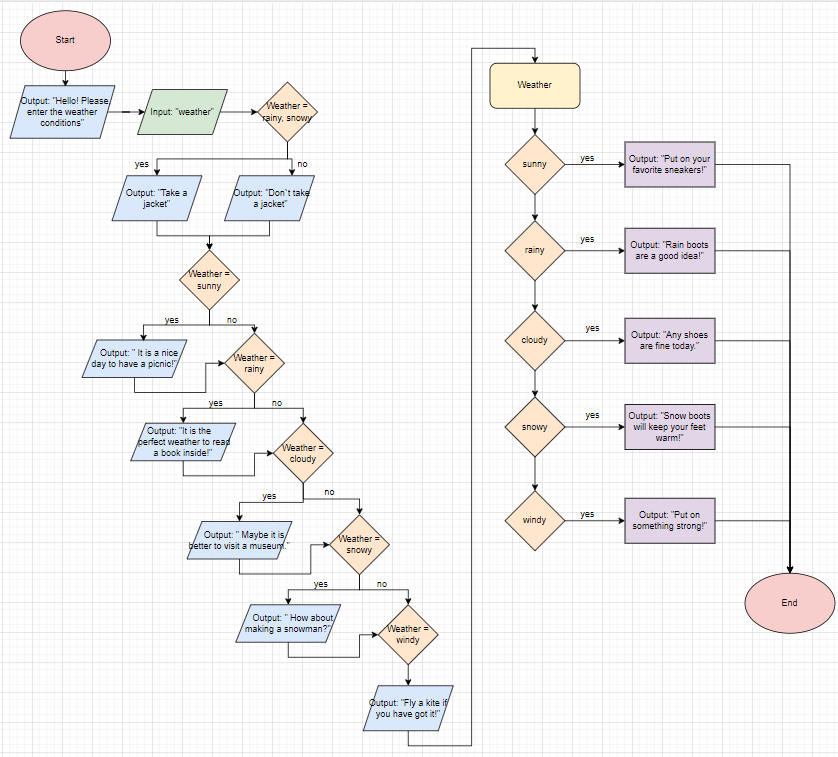


Рисунок 1: Блок-схема до програми №1

* Планований час на реалізацію: 1,5 год

Програма №2 “VNS\_LAB\_1\_1”

* Блок-схема

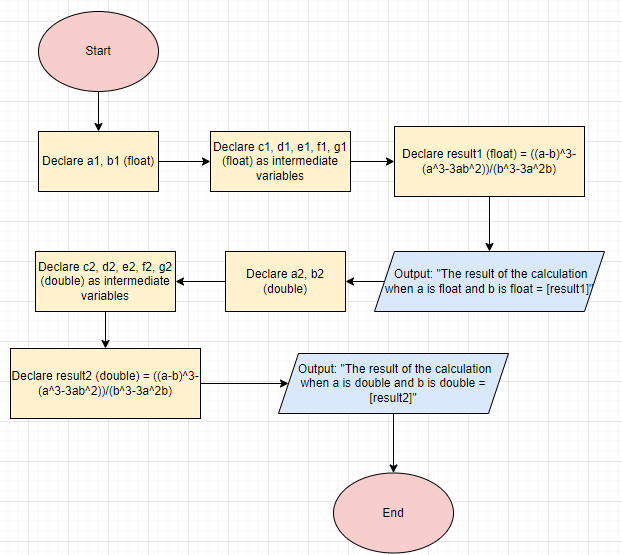


Рисунок 2: Блок-схема до програми №2

Планований час на реалізацію: 50 хв

Програма №3 “VNS\_LAB\_1\_2”

* Блок-схема

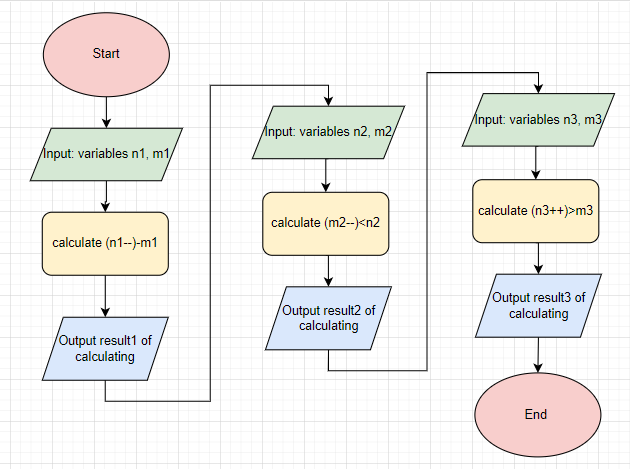


Рисунок 3: Блок-схема до програми №3

* Планований час на реалізацію: 35 хв

Програма №4 “VNS\_LAB\_2”

* Блок-схема

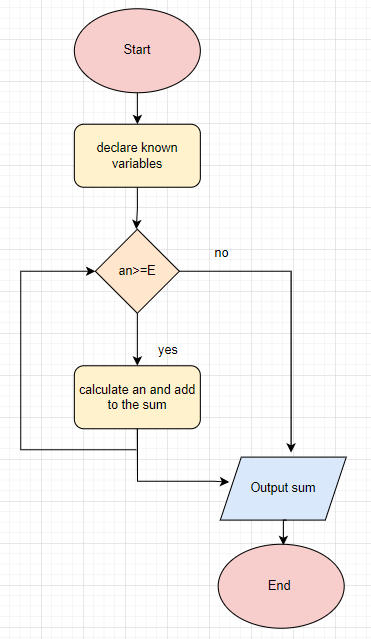


Рисунок 4: Блок-схема до програми №4

* Планований час на реалізацію: 45 хв

Програма №5 “Algotester\_lab1”

* Блок-схема

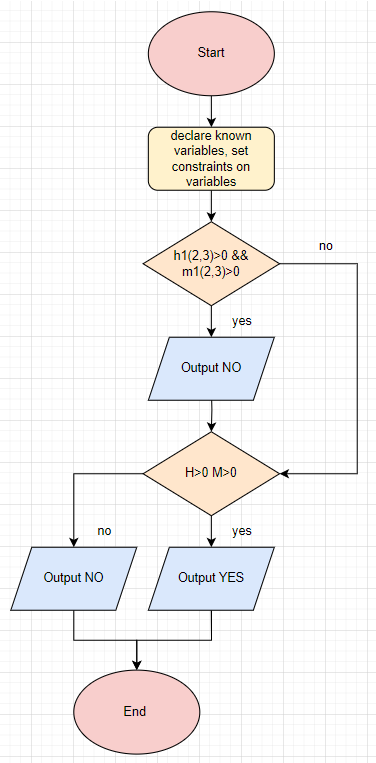


Рисунок 5: Блок-схема до програми №5

* Планований час на реалізацію: 1,2 год

Програма №6 “Algotester\_lab2”

Блок-схема

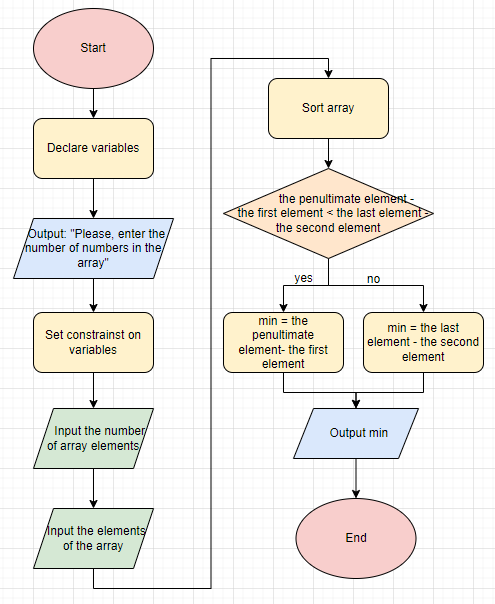


Рисунок 6: Блок-схема до програми №6

* Планований час на реалізацію: 1,5 год
* Програма №7 “Self\_Practice\_Task”
* Блок-схема

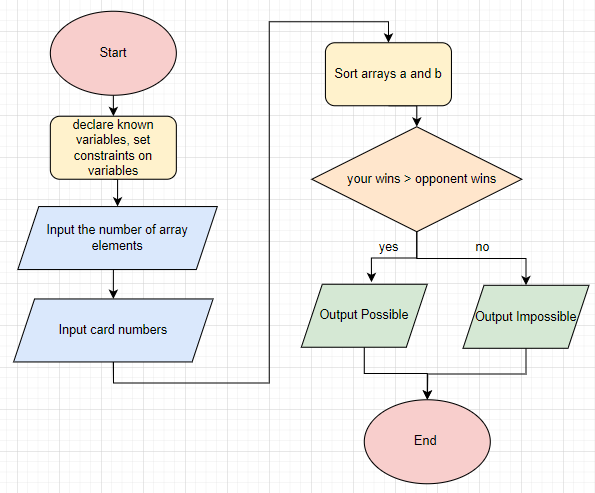


Рисунок 7: Блок-схема до програми №7

* Планований час на реалізацію: 2 год

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

Додаткова конфігурація середовища до виконання завдань не потрібна.

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

Завдання №1 ”Practice\_Epic\_2”

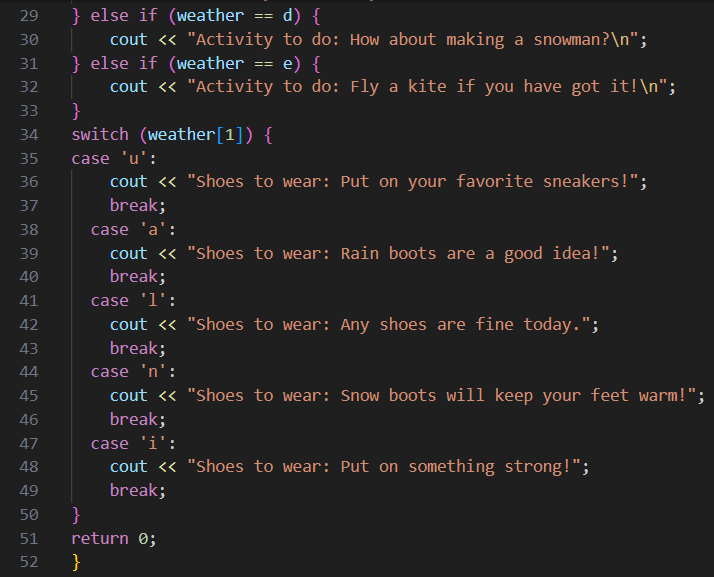
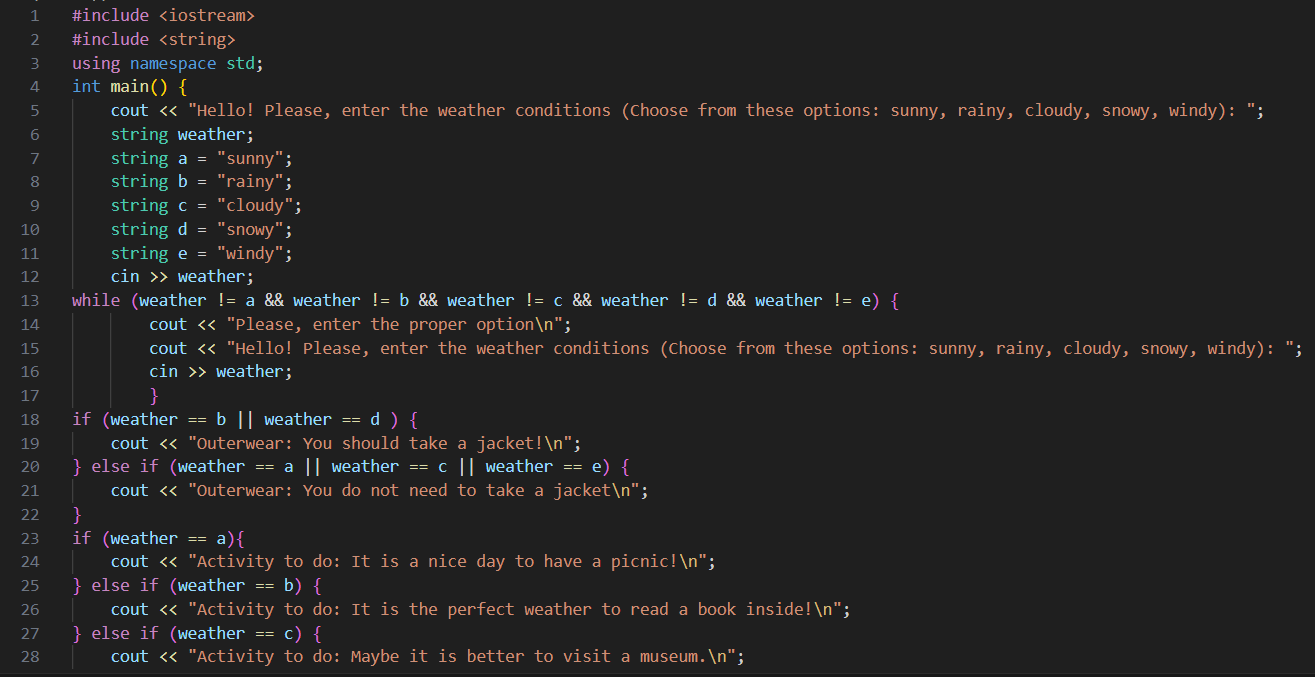


Рисунок 8: Код до програми №1

Деталі завдання: Створено простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди. Запропонувано користувачеві ввести поточні погодні умови: sunny; rainy; cloudy; snowy; windy. Якщо користувач вводить будь-яку іншу умову, запропонувано йому ввести дійсну умову. Якщо йде сніг або дощ, користувач повинен одягнути куртку. В іншому випадку куртка не потрібна. Якщо сонячно, порекомендувано «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, порекомендувано «Як щодо того, щоб зліпити сніговика?». Інакше, якщо буде вітер, порекомендувано «Запустіть повітряного змія, якщо він у вас є!». Рекомендації щодо взуття: 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!"

Важливі деталі для врахування в імплементації програми: Перевірено значення погоди (воно має бути одним з запропонованих варіантів: sunny; rainy; cloudy; snowy; windy.

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/ae6b99d72bb22a33c2b2c93d7e67743a3bd415e3/ai_12/mariia_brychko/Epic_2/prac_2.cpp>

Завдання №2 “VNS\_LAB\_1\_1”

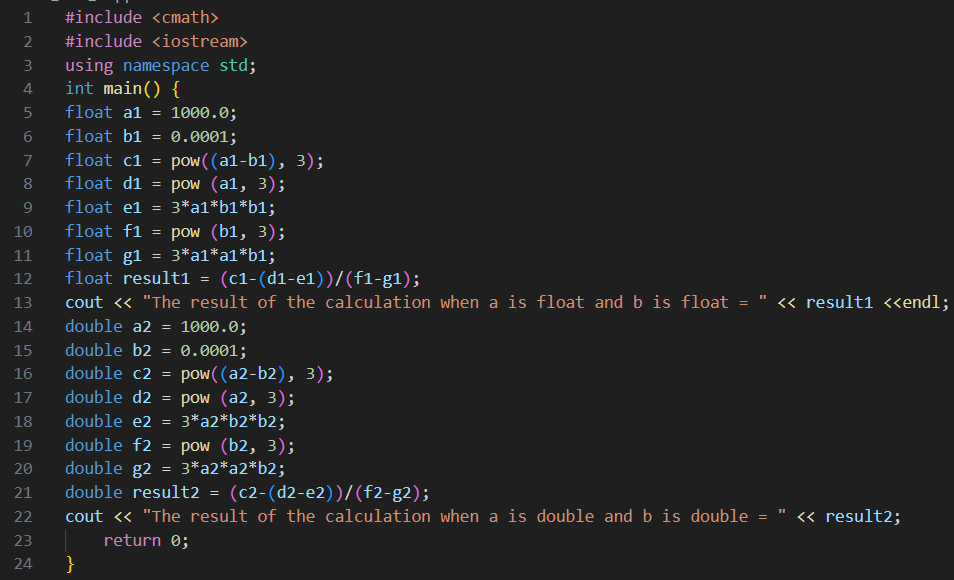


Рисунок : Код до програми №2

Деталі завдання: Обчислено значення виразу за формулою при різних дійсних типах даних (float й double). Обчислення виконано з використанням проміжних змінних. Виведено результат.

Важливі деталі для врахування в імплементації програми: Перевірено чи всі змінні у першому випадку мають тип даних float, і у другому - double.

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/ae6b99d72bb22a33c2b2c93d7e67743a3bd415e3/ai_12/mariia_brychko/Epic_2/vns_lab1_1.cpp>

Завдання №3 “VNS\_LAB\_1\_2”

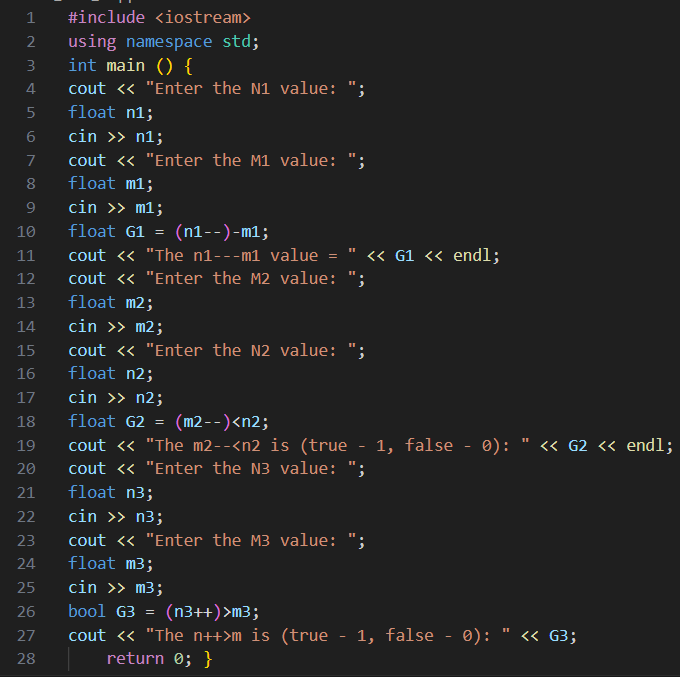


Рисунок : Код до програми №3

Деталі завдання: Обчислено значення заданих виразів.

Важливі деталі для врахування в імплементації програми: Важливо, що ми маємо саме постфіксний інкремент/декремент.

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/ae6b99d72bb22a33c2b2c93d7e67743a3bd415e3/ai_12/mariia_brychko/Epic_2/vns_lab1_2.cpp>

Завдання №4 “VNS\_LAB\_2”

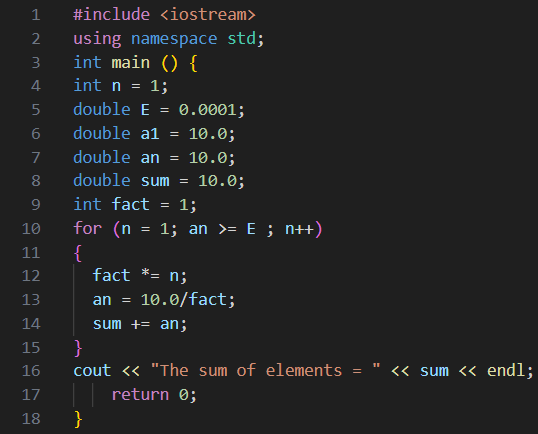


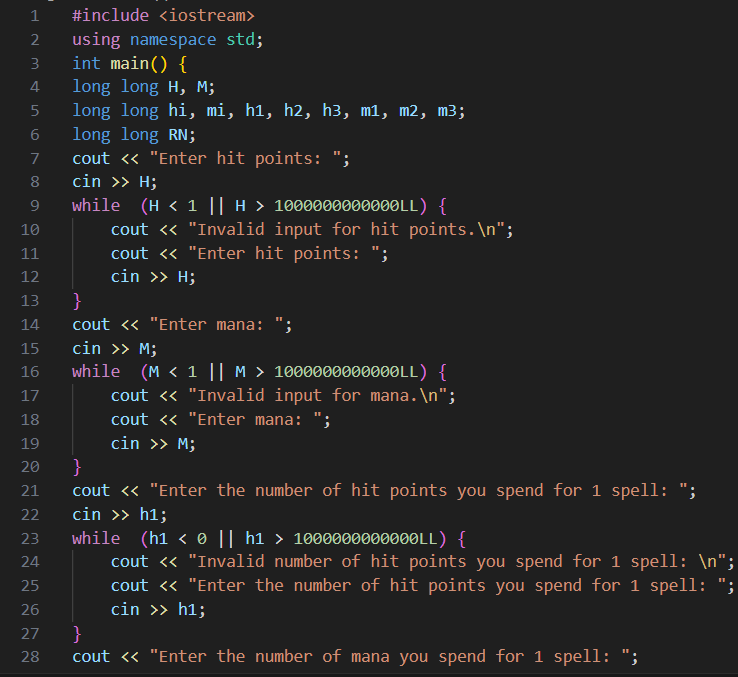
Рисунок : Код до програми №4

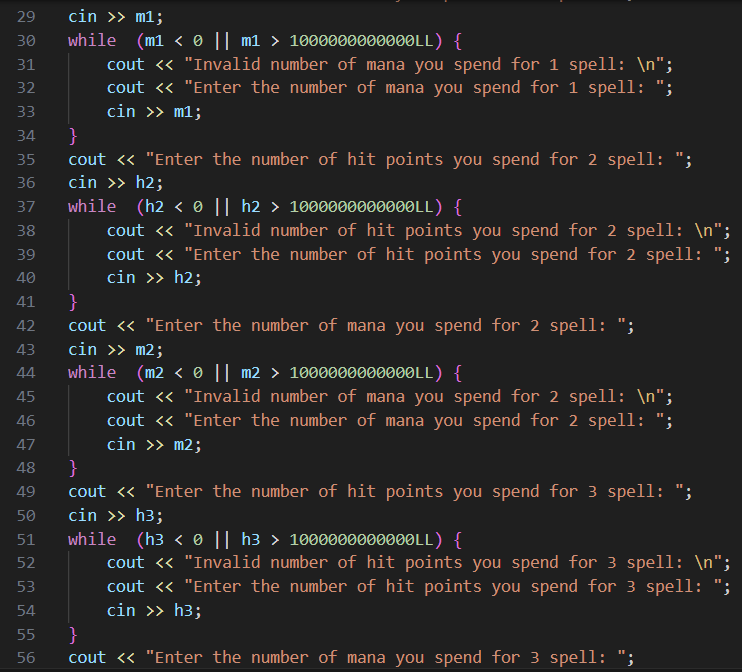
Деталі завдання: Використовуючи оператор циклу, знайдено суму ряду з точністю ε=0.0001, за даною формулою. Виведено результат.

Важливі деталі для врахування в імплементації програми: При визначенні суми членів ряду знайдено рекурентну формулу для отримання наступного члена ряду.

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/ae6b99d72bb22a33c2b2c93d7e67743a3bd415e3/ai_12/mariia_brychko/Epic_2/vns_lab_2.cpp>

Завдання №5 “Algotester\_lab1”





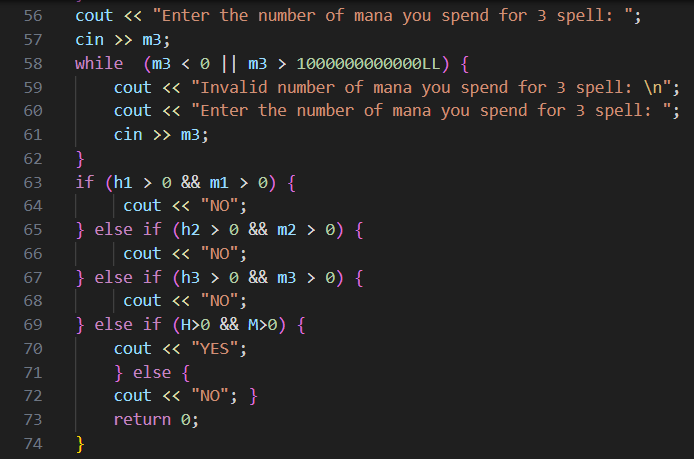


Рисунок 12: Код до програми №5

Деталі завдання: У персонажа є H хiтпойнтiв та M мани. Персонаж 3 рази використає закляття, кожне з яких може використати хiтпойнти та ману одночасно. Якщо якесь закляття забирає i хiтпойнти i ману - персонаж програє. Якщо в кiнцi персонаж буде мати додатню кiлькiсть хiтпойнтiв та мани – вiн виграє, в iншому випадку програє. У випадку виграшу персонажа вивести YES, вивести NO у iншому випадку.

Важливі деталі для врахування в імплементації програми: Для виграшу використано при одному заклинаннi АБО хiтпойнти, АБО ману. Також враховано наступні обмеження:

1 ≤ H ≤ 1012

1 ≤ M ≤ 1012

0 ≤ hi ≤ 1012

0 ≤ mi ≤ 1012

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/ae6b99d72bb22a33c2b2c93d7e67743a3bd415e3/ai_12/mariia_brychko/Epic_2/algotester_lab1.cpp>

Завдання №6 “Algotester\_lab2”

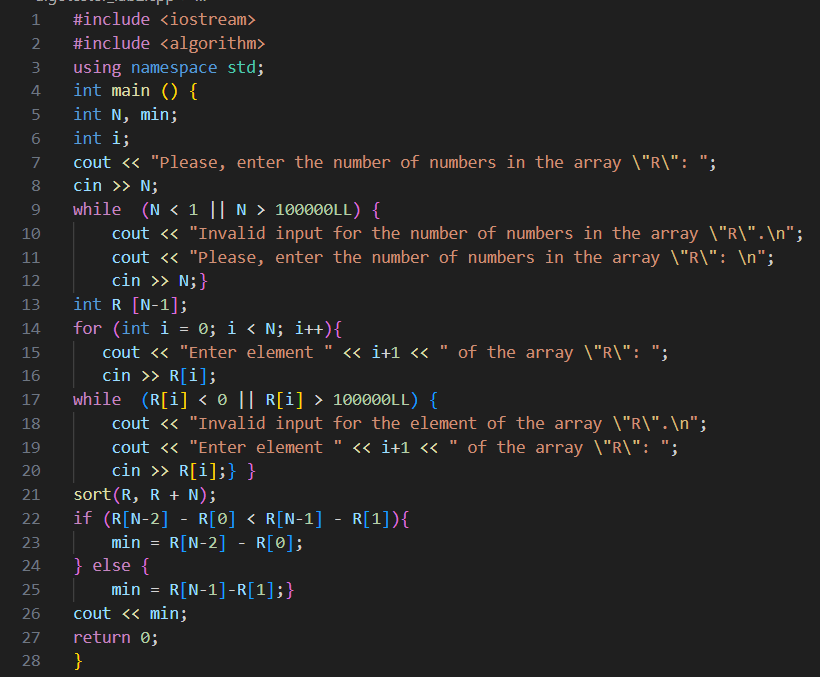


Рисунок Рисунок 12: Код до програми №6

Деталі завдання: Є дорога, яка виглядає як N чисел. Пiсля того як по нiй проїхали - втому визначаємо як рiзницю максимального та мiнiмального елементу. Ми мiнiмiзовуємо втому шляхом викидання одного числа з дороги, тобто забиранням його з масиву. В результатi цiєї дiї, обчислено яку мiнiмальну втому отримуємо в кiнцi дороги.

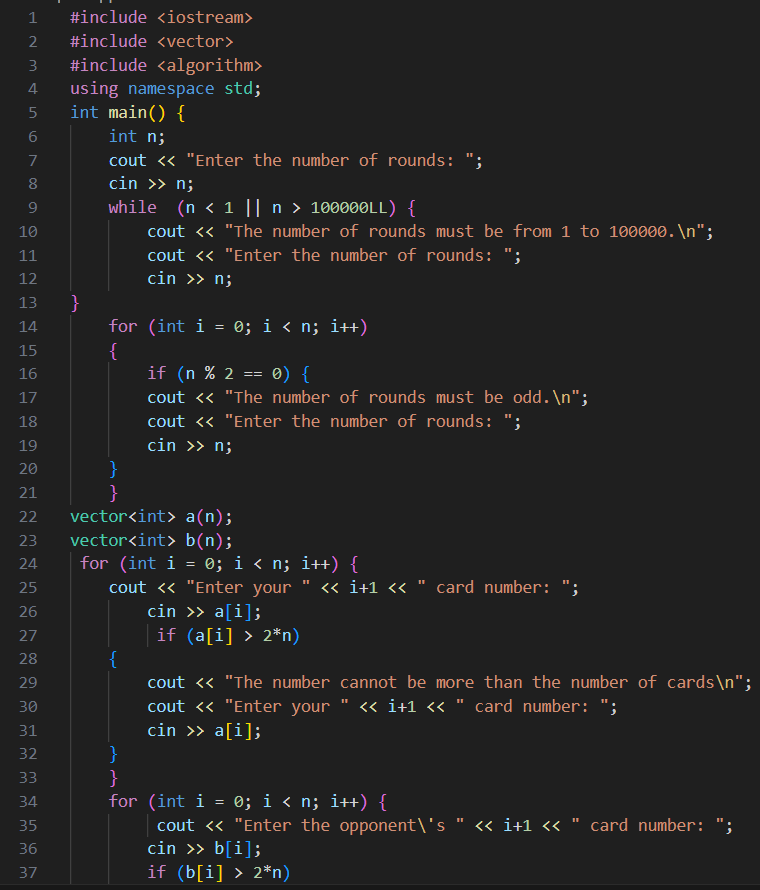
Важливі деталі для врахування в імплементації програми: Враховано наступні обмеження:

1 ≤ N ≤ 105

0 ≤ ri ≤ 105

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/ae6b99d72bb22a33c2b2c93d7e67743a3bd415e3/ai_12/mariia_brychko/Epic_2/algotester_lab2.cpp>

Завдання №7 “Self\_Practice\_Task”



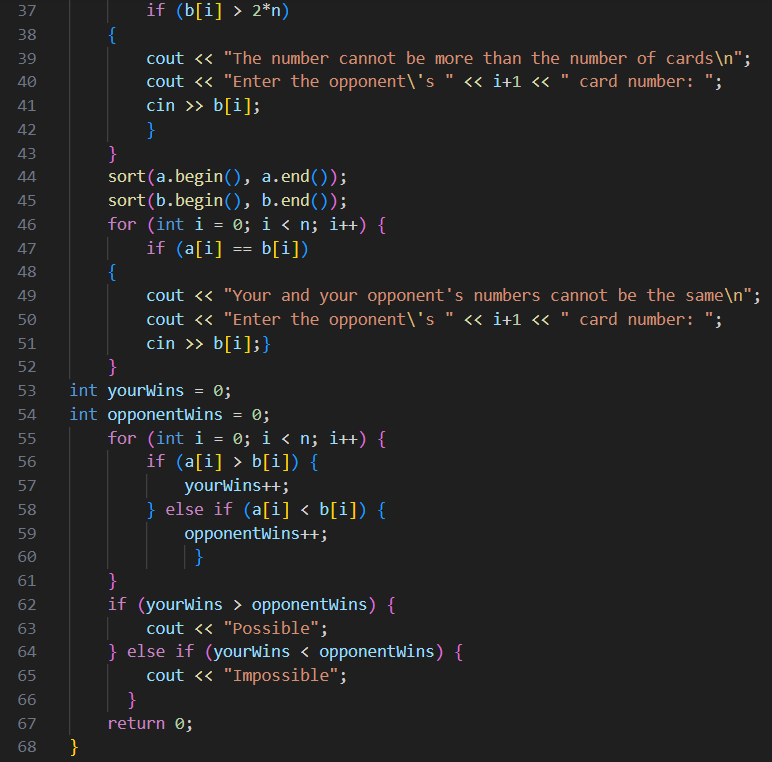


Рисунок Рисунок 12: Код до програми №7

Деталі завдання: В колодi знаходиться 2 · n карт. На кожнiй картi написано число вiд 1 до 2 · n. Числа на всiх картах рiзнi. В гру грають двоє гравцiв. На початку гри гравцi перемiшують карти в колодi i кожен з гравцiв отримує n карт. Гра складається з n раундiв. В кожному раундi гравцi обирають по однiй зi своїх ще не зiграних карт i одночасно кладуть на стiл. Раунд виграє той гравець, який поставив карту з бiльшим числом. Переможець гри — той, хто виграв бiльшу кiлькiсть раундiв. Гравцi уже перемiшали колоду i роздали карти. Нам добре відомо стратегiю свого суперника — послiдовнiсть карт, якими вiн ходитиме. Визначено чи є можливiсть перемогти у грi. Виведено слово Possible, якщо можливо перемогти, або Impossible, якщо неможливо.

Важливі деталі для врахування в імплементації програми: Враховано наступні обмеження:

1 ≤ n ≤ 105,

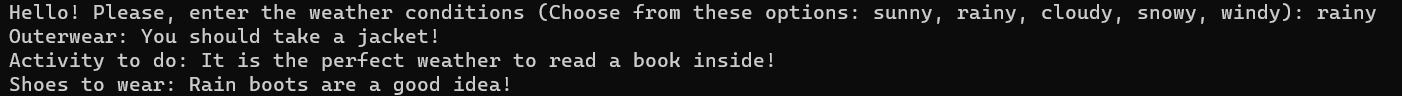
n — непарне,

об’єднання масивiв a i b формує перестановку чисел вiд 1 до 2 · n.

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/ae6b99d72bb22a33c2b2c93d7e67743a3bd415e3/ai_12/mariia_brychko/Epic_2/selfprac.cpp>

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

Завдання №1 ”Practice\_Epic\_2”

 У консоль уводяться вхідні дані – погода. На скріншоті показано приклад виводу даних з консолі коли погода дощова.

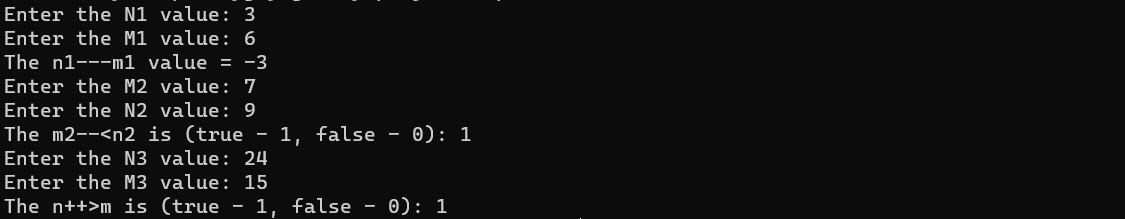
Час затрачений на виконання завдання: 1 год

Завдання №2 “VNS\_LAB\_1\_1”



Після компіляції програми у консоль виводяться результати обчислень коли a i b – float, a i b – double.

Час затрачений на виконання завдання: 30 хв

Завдання №3 “VNS\_LAB\_1\_2” 

У консоль уводяться вхідні дані – значення n1, m1, n2, m2, n3, m3. На скріншоті показано приклад виводу даних з консолі.

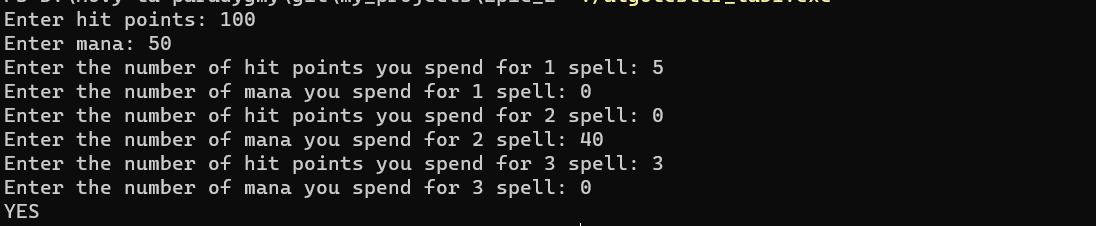
Час затрачений на виконання завдання: 30 хв

Завдання №4 “VNS\_LAB\_2” 

Після компіляції програми у консоль виводиться обчислення суми ряду з точністю ε=0.0001, за даною формулою.

Час затрачений на виконання завдання: 30 хв

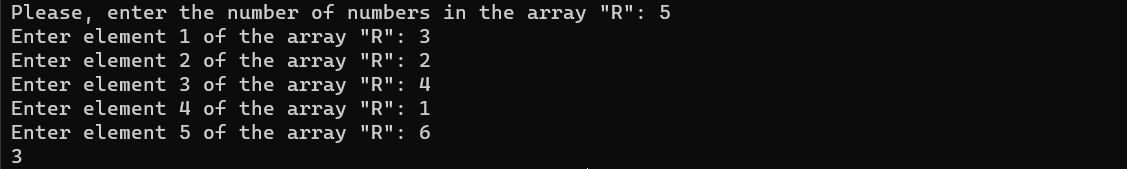
Завдання №5 “Algotester\_lab1”



У консоль уводяться вхідні дані – значення H, M, h1, m1, h2, m2, h3, m3. На скріншоті показано приклад виводу даних з консолі(коли всі умові виконуються).

Час затрачений на виконання завдання: 1,5 год

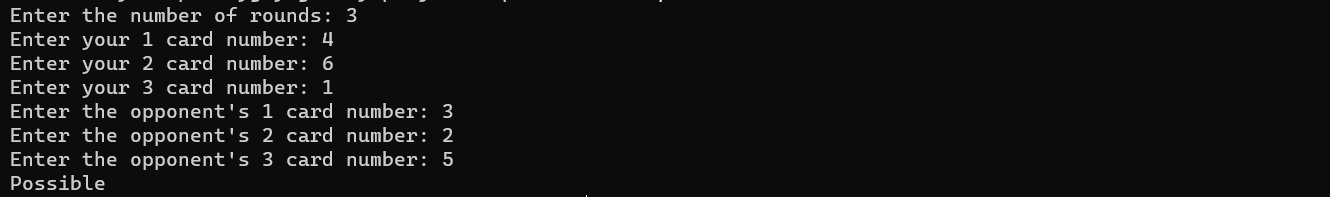
Завдання №6 “Algotester\_lab2”



У консоль уводяться вхідні дані – довжина масиву і значення елементів масиву. На скріншоті показано приклад виводу даних з консолі. Втома мінімізована.

Час затрачений на виконання завдання: 1,5 год

Завдання №7 “Self\_Practice\_Task”



У консоль уводяться вхідні дані – кількість раундів і числа на картах. На скріншоті показано приклад вхідних даних – і як результат у консолі виводиться Possible.

Час затрачений на виконання завдання: 4 год

# **Висновки:**

Написано лабораторні роботи на мові С++ у VS Code. Створено блок-схеми до коду. Застосовано теоретичні знання з епіку 1 на практиці. Глибше вивчено мову С++: константи, змінні, лінійні та розгалужені алгоритми, умовні оператори, цикли