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 “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 №5 “Self\_Practice\_Task\_1”

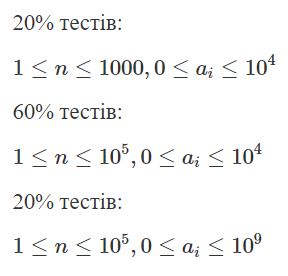
* Варіант завдання: без варіанту
* Деталі завдання: Є депутат, у якого є необмежена кількість купюр усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Він хоче придбати подарунок для своєї тещі, що коштує n гривень. Він хоче оплатити покупку готівкою без решти. Вивести число, яка мінімальна кількість купюр йому для цього знадобиться.
* Важливі деталі для врахування в імплементації програми: Необхідно врахувати наступні обмеження:
* 1≤n≤10^9.

Завдання №6 “Self\_Practice\_Task\_2”

* Варіант завдання: без варіанту
* Деталі завдання: Марічка дуже любить печиво. Сьогодні, коли всі пластуни покинуть свої намети і підуть купатися в річку, Марічка планує непомітно з’їсти трохи печива. Звісно, Марічка не хоче бути покараною і дуже боїться, щоб Зеник не помітив пропажу.

Марічка підгледіла, скільки пачок печива є в рюкзаку Зеника. Також вона знає, скільки штук в кожній пачці. Марічці не терпиться дізнатися, скільки ж печива вона зможе з’їсти так, щоб Зеник не помітив. Зеник помітить пропажу печива з деякої пачки тоді і тільки тоді, коли Марічка повністю спустошить її. Вивести одне ціле число — максимальну кількість штук печива, яку зможе з’їсти Марічка так, щоб Зеник не помітив цього.

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

* 

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

Програма №1 ”Practice\_Epic\_2”

* Блок-схема

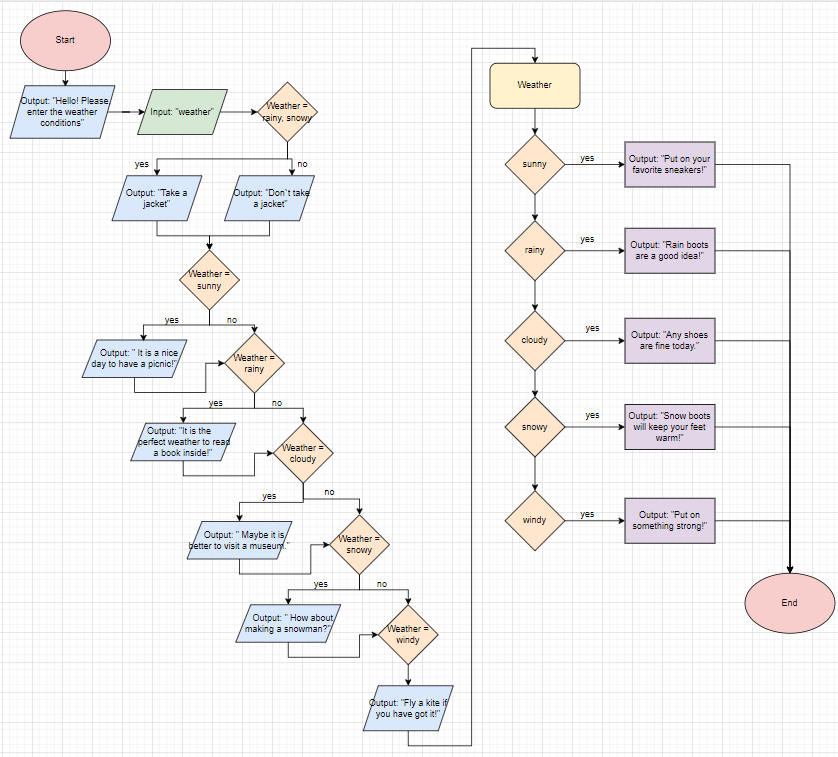


Рисунок 1: Блок-схема до програми №1

* Планований час на реалізацію: 1,5 год

Програма №2 “VNS\_LAB\_1\_1”

* Блок-схема

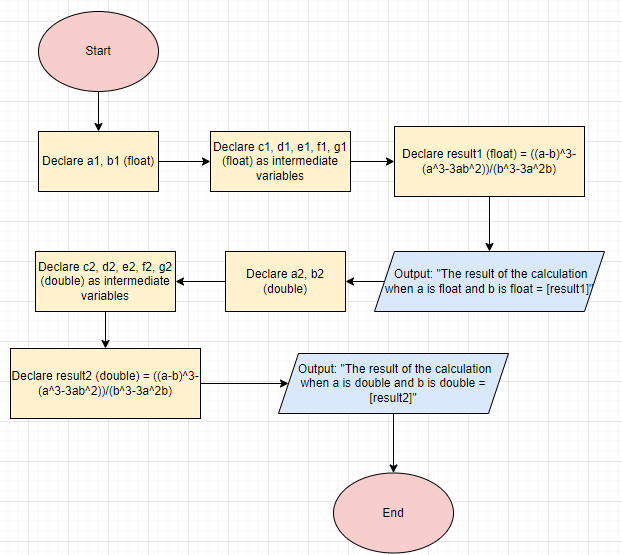


Рисунок 2: Блок-схема до програми №2

Планований час на реалізацію: 50 хв

Програма №3 “VNS\_LAB\_1\_2”

* Блок-схема

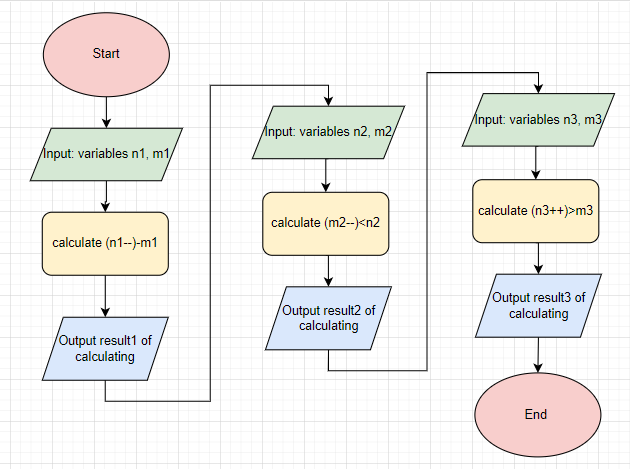


Рисунок 3: Блок-схема до програми №3

* Планований час на реалізацію: 35 хв

Програма №4 “Algotester\_lab1”

* Блок-схема

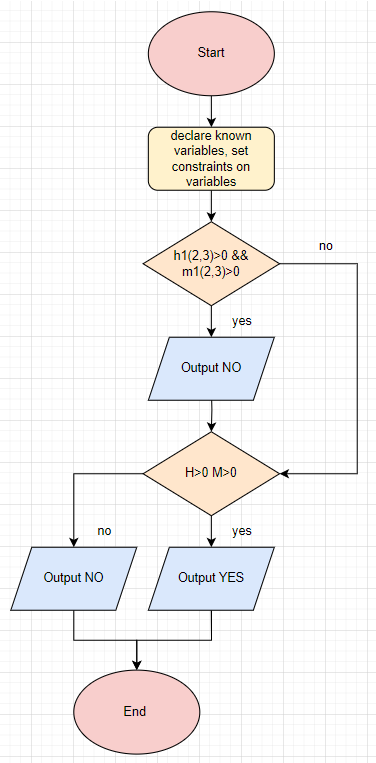


Рисунок 4: Блок-схема до програми №4

* Планований час на реалізацію: 1 год
* Програма №5 “Self\_Practice\_Task\_1”
* Блок-схема

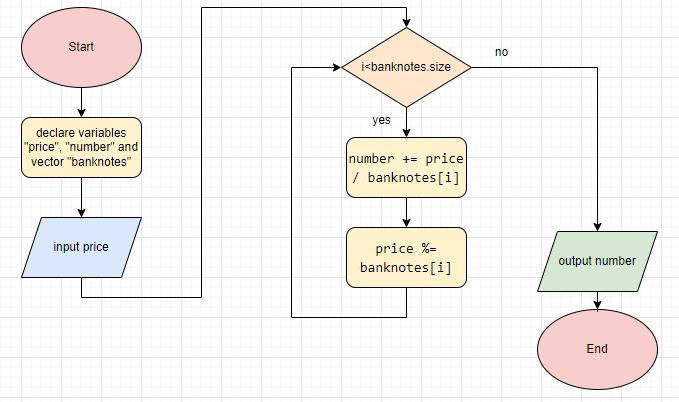


Рисунок 5: Блок-схема до програми №5

* Планований час на реалізацію: 30 хв
* Програма №6 “Self\_Practice\_Task\_2”
* Блок-схема

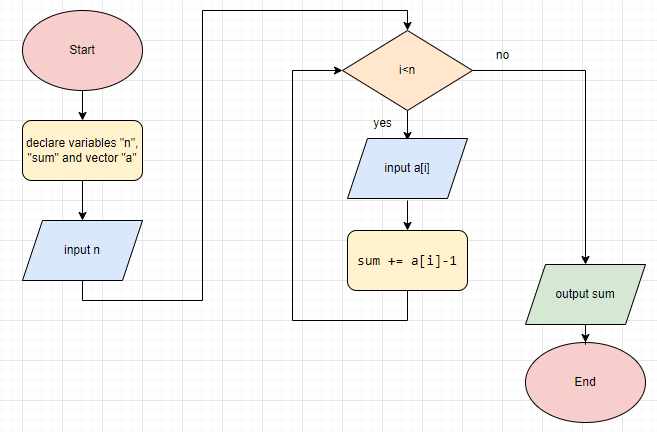


Рисунок 6: Блок-схема до програми №6

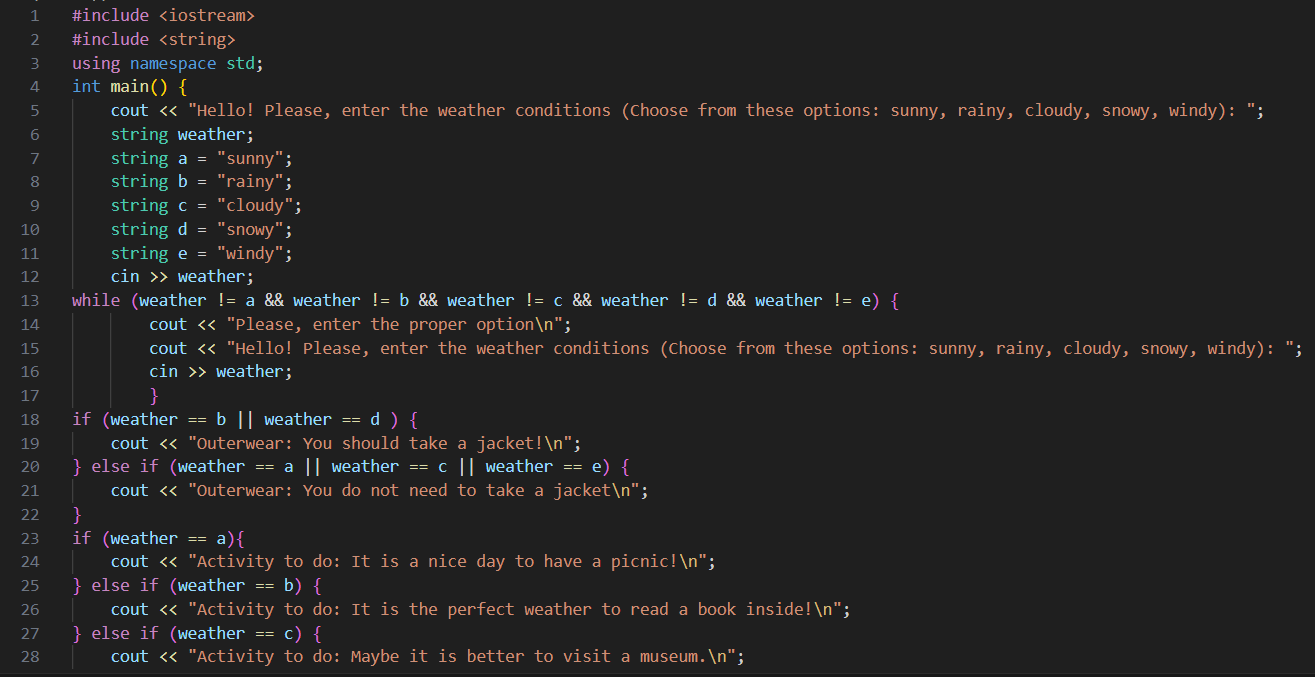
* Планований час на реалізацію: 30 хв

## **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”



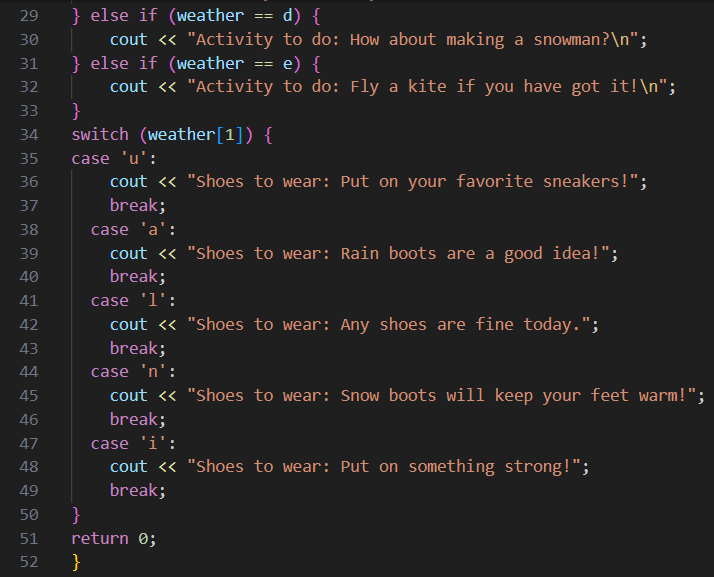


Рисунок 7: Код до програми №1

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/d799702b0952887d4763e96899bc7e7894f6b67b/ai_12/mariia_brychko/Epic_2/practice_work_team_tasks_maria_brychko.cpp.cpp>

Завдання №2 “VNS\_LAB\_1\_1”

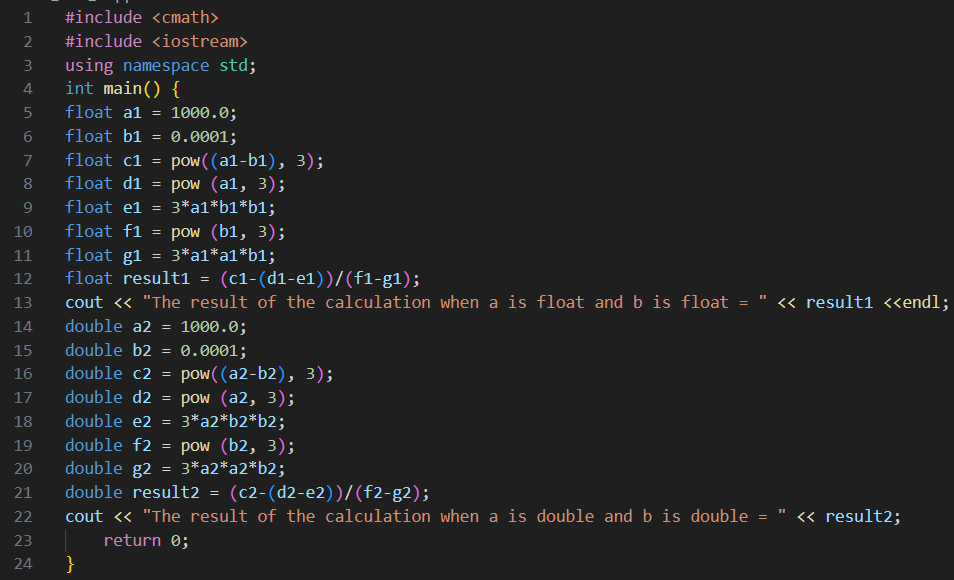


Рисунок 8: Код до програми №2

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/d799702b0952887d4763e96899bc7e7894f6b67b/ai_12/mariia_brychko/Epic_2/vns_lab_1_task_1_1_maria_brychko.cpp.cpp>

Завдання №3 “VNS\_LAB\_1\_2”

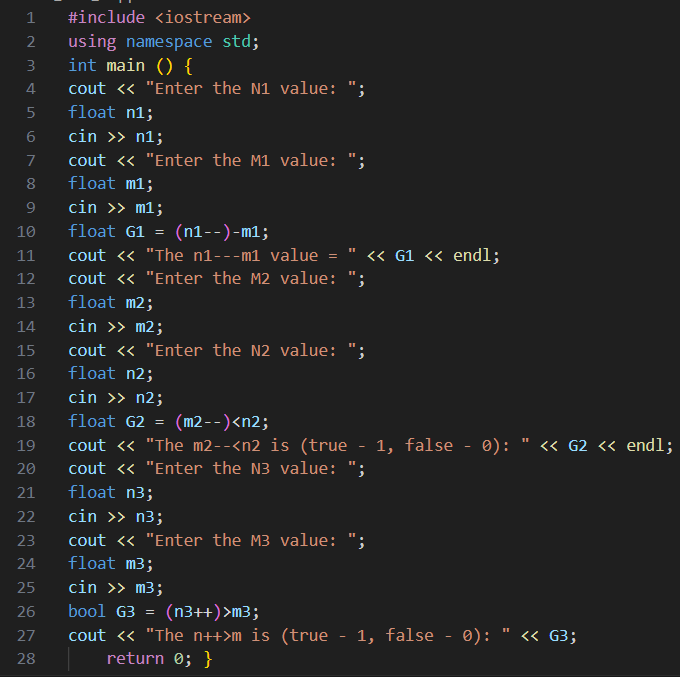


Рисунок 9: Код до програми №3

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/d799702b0952887d4763e96899bc7e7894f6b67b/ai_12/mariia_brychko/Epic_2/vns_lab_1_task_1_2_maria_brychko.cpp.cpp>

Завдання №4 “Algotester\_lab1”

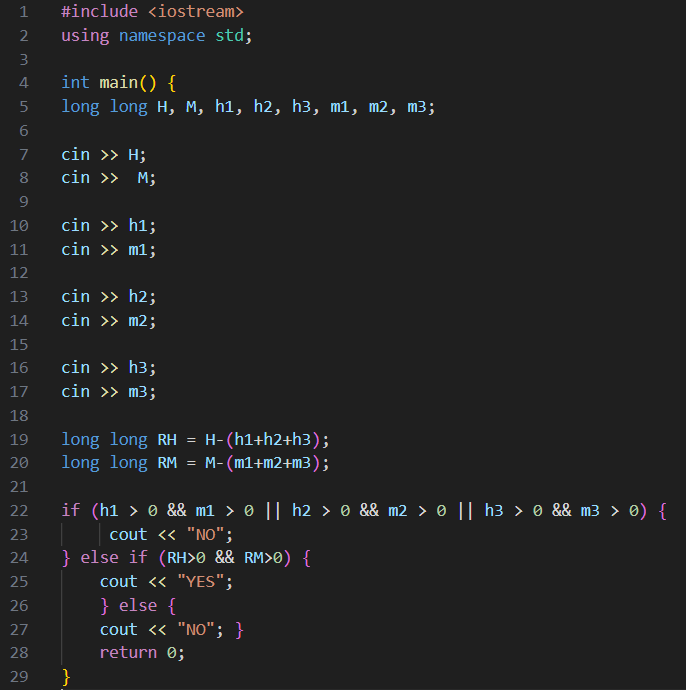


Рисунок 10: Код до програми №4

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 виводиться YES, виводиться 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/d799702b0952887d4763e96899bc7e7894f6b67b/ai_12/mariia_brychko/Epic_2/algotester_lab_1_task_1_maria_brychko.cpp.cpp>

Завдання №5 “Self\_Practice\_Task\_1”

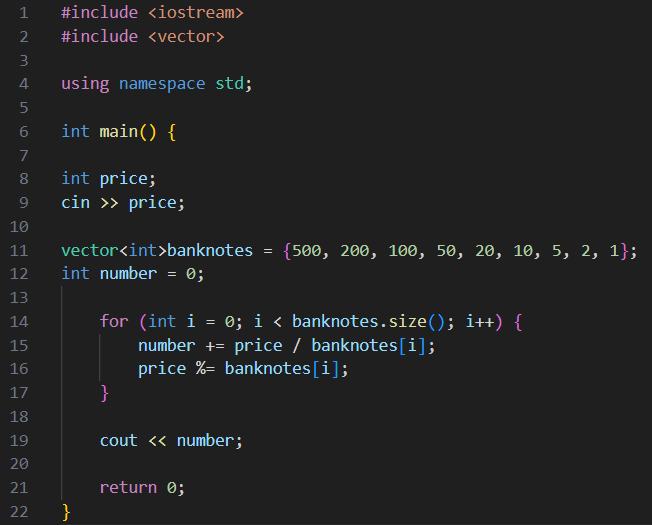


Рисунок 11: Код до програми №5

* Деталі завдання: Створено масив, у якого елементи це купюри усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Створено цикл, у якому підраховується кількість потрібних купюр, цикл триває до останнього елемента масиву. Виводиться число, яка мінімальна кількість купюр потрібна.
* Важливі деталі для врахування в імплементації програми: Враховано наступні обмеження:
* 1≤n≤10^9.

Посилання на файл програми у пул-запиті GitHub <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/d799702b0952887d4763e96899bc7e7894f6b67b/ai_12/mariia_brychko/Epic_2/practice_work_self_algotester_task_1_maria_brychko.cpp.cpp>

Завдання №6 “Self\_Practice\_Task\_2”

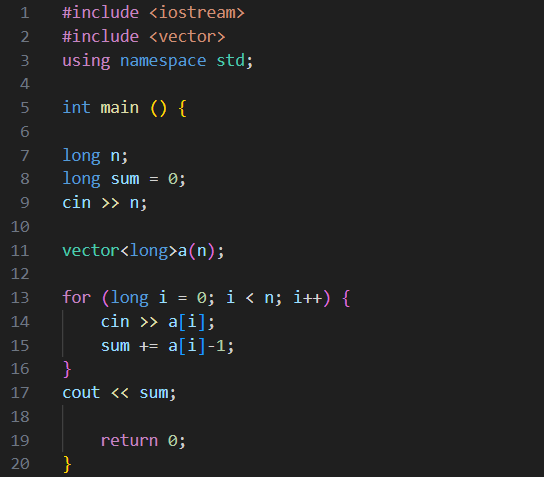
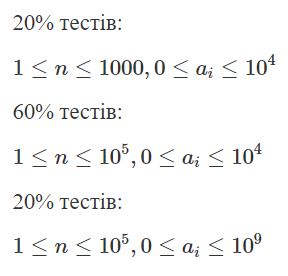


Рисунок 12: Код до програми №6

* Деталі завдання: Програма підраховує суму усього печива, яке може з’їсти Марічка, за допомогою циклу. Виводиться одне ціле число — максимальну кількість штук печива, яку зможе з’їсти Марічка так, щоб Зеник не помітив цього.

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



Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/d799702b0952887d4763e96899bc7e7894f6b67b/ai_12/mariia_brychko/Epic_2/practice_work_self_algotester_task_2_maria_brychko.cpp>

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

Завдання №1 ”Practice\_Epic\_2”

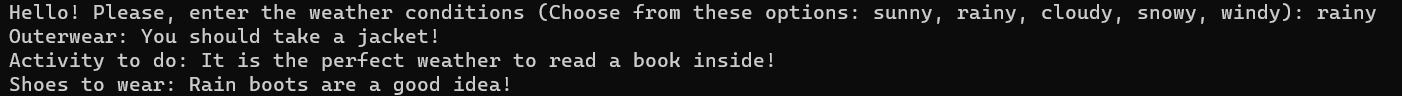


Рисунок 13: Результати програми №1

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”



Рисунок 14: Результати програми №2

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”

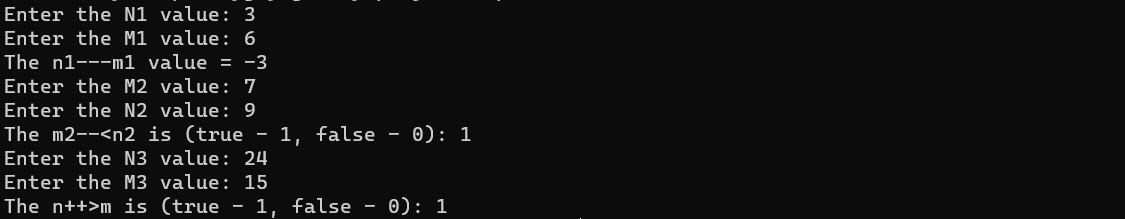


Рисунок 15: Результати програми №3

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 “Algotester\_lab1”

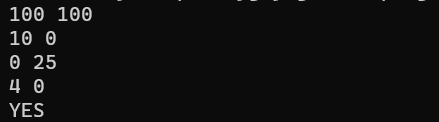


Рисунок 16: Результати програми №4

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 год

Завдання №5 “Self\_Practice\_Task\_1” 

Рисунок 17: Результати програми №5

У консоль уводяться вхідні дані – сума подарунку. На скріншоті показано приклад вхідних даних – і як результат у консолі виводиться необхідна кількість купюр.

Час затрачений на виконання завдання: 30 хв

Завдання №6 “Self\_Practice\_Task\_2”



Рисунок 18: Результати програми №6

У консоль уводяться вхідні дані – кількість пачок печива і кількість печива у кожній пачці. На скріншоті показано приклад вхідних даних – і як результат у консолі виводиться найбільша можлива кількість печива, яку Марічка може з’їсти.

Час затрачений на виконання завдання: 30 хв

# **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