Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**про виконання лабораторних та практичних робіт блоку № 2**

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

***з дисципліни:*** «Основи програмування»

до:

ВНС Лабораторної Роботи № 1

Алготестер Лабораторної Роботи № 1

Практичних Робіт до блоку № 2

**Виконав:**

Студент групи ШІ-13

Бойко Роман Андрійович

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

**Мета роботи:** Навчитися використовувати логічні оператори для різних видів завдань, розібратися з різними типами даними та вбудованими функціями.

**Теоретичні відомості:**

* умовні оператори if та switch
* ввід та вивід даних на мовах C/C++
* типи даних

Джерела

* <https://itproger.com/ua/course/cpp/4>
* <https://www.programiz.com/c-programming/c-input-output>
* <https://www.scholarhat.com/tutorial/cpp/data-types-in-cpp>
* Конспект лекцій з програмування О. Ю. Пшеничний

### **Виконання роботи**

**Завдання 1: VNS Lab 1**

**Особистий варіант - 25**

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

Обчислення варто виконувати з використанням проміжних змінних.

Порівняти й пояснити отримані результати.

при а=1000, b=0.0001

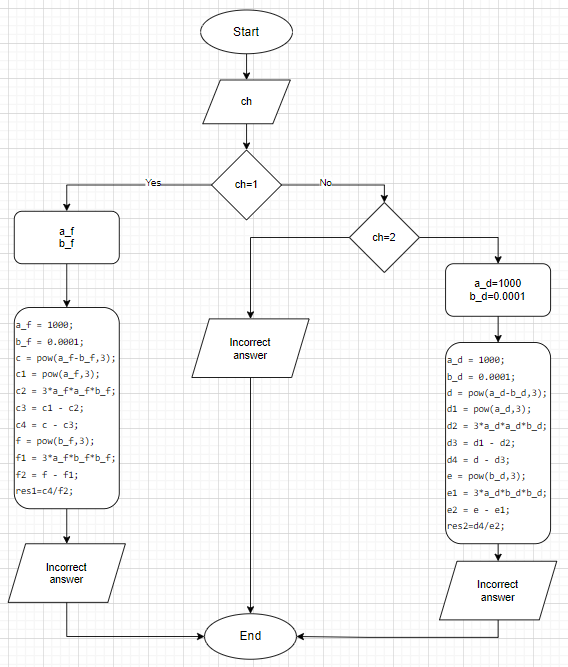
**2.** Обчислити значення виразів. Пояснити отримані результати.

1) - -m-++n

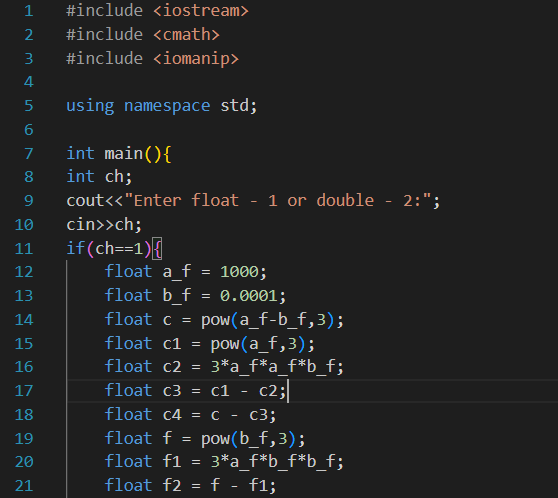
2) m\*n<n++

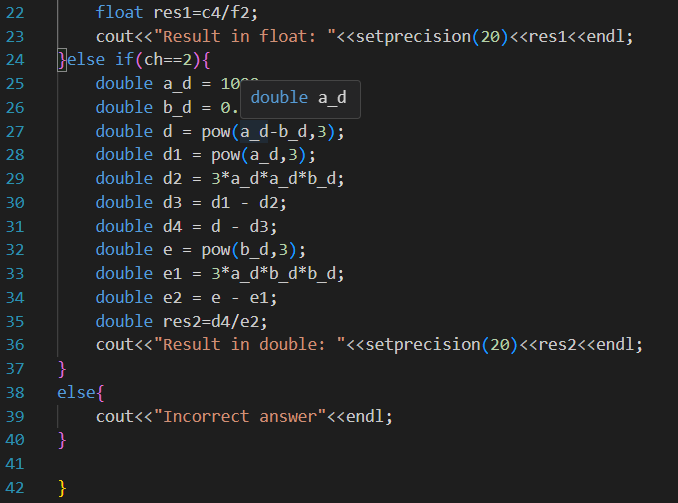
3) n-- > m++

**Блок-схема до задачі:**

****

**Розв’язок задачі**

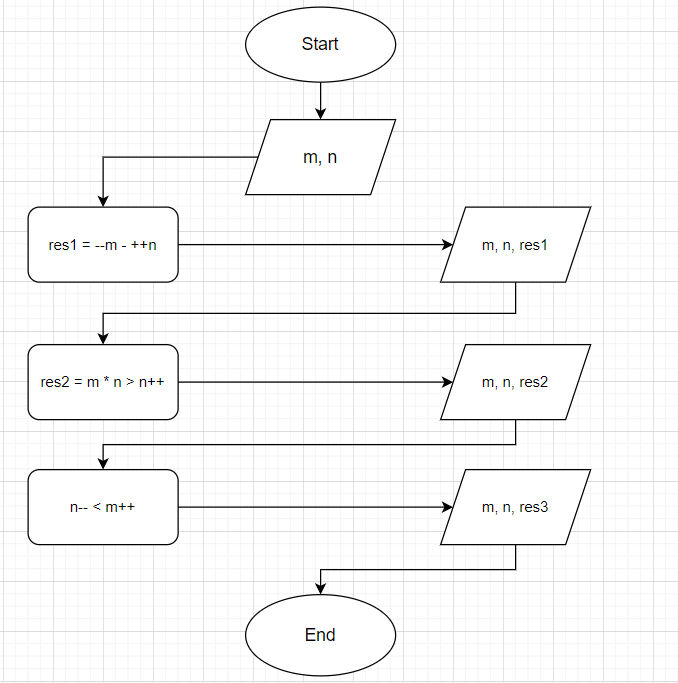
****

****

**Результат виконання програми:**

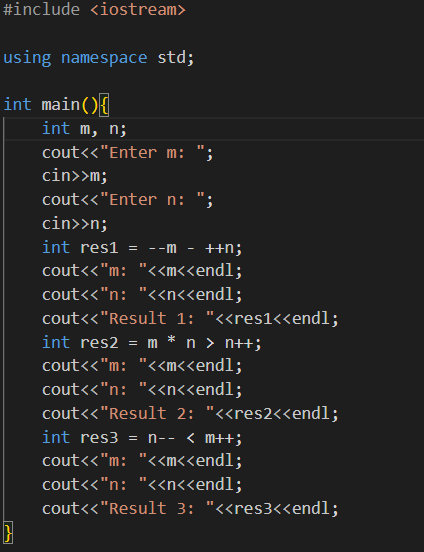
****

**Час виконання завдання ~ 20 хв**

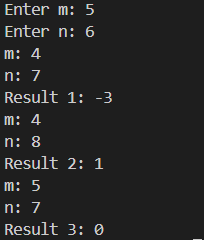
**2)**

**Блок-схема до задачі:**

**Розв’язок задачі:**

****

**Результат виконання програми:**

****

**Час виконання завдання ~ 10 хв**

**Завдання 2: Algotester Lab 1**

**Особистий варіант - 3**

Персонажу по одному дають сторони 5 кубів a1..5, з яких він будує піраміду.

Коли він отримує куб з ребром ai - він його ставить на існуючий, перший ставить на підлогу (вона безмежна).

Якщо в якийсь момент об’єм куба у руці (який будуть ставити) буде більший ніж у куба на вершині піраміди - персонаж програє і гра закінчується. Розмір усіх наступних кубів після програшу не враховується.

Тобто якщо ai−1<ai - це програш.

Ваше завдання - сказати як закінчиться гра.

# **Вхідні дані**

5 цілих чисел

a1..5 - сторони кубів

# **Вихідні дані**

Існуючі варіанти:

**LOSS** - якщо персонаж не зможе поставити куб.

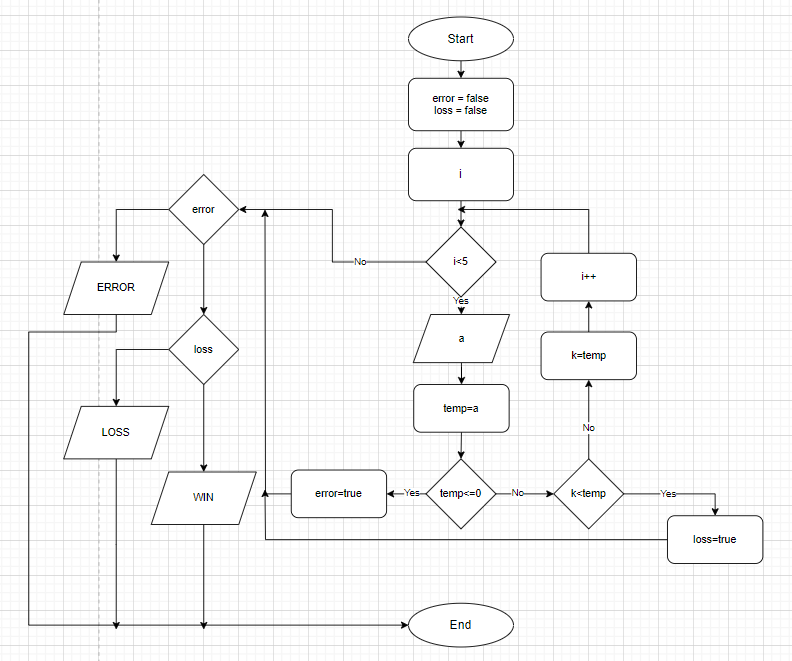
**WIN** - якщо персонаж зможе поставити усі куби

**ERROR** - якщо сторона куба буде мати неможливу довжину, тобто

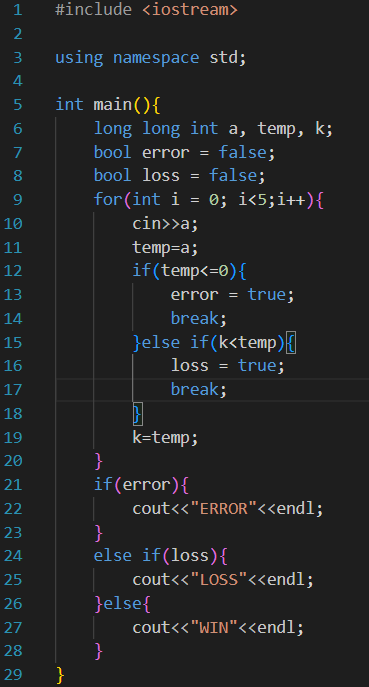
ai<= 0

# **Обмеження**

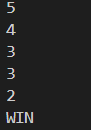
−10^12≤ai≤10^12

**Блок-схема до задачі:**

**Розв’язок задачі:**

****

**Результат виконання програми:**

****

**Час виконання завдання ~ 20 хв**

**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;

## Мета Задачі

Навчитися користуватися операторами галуження для структурування логіки програм.

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)

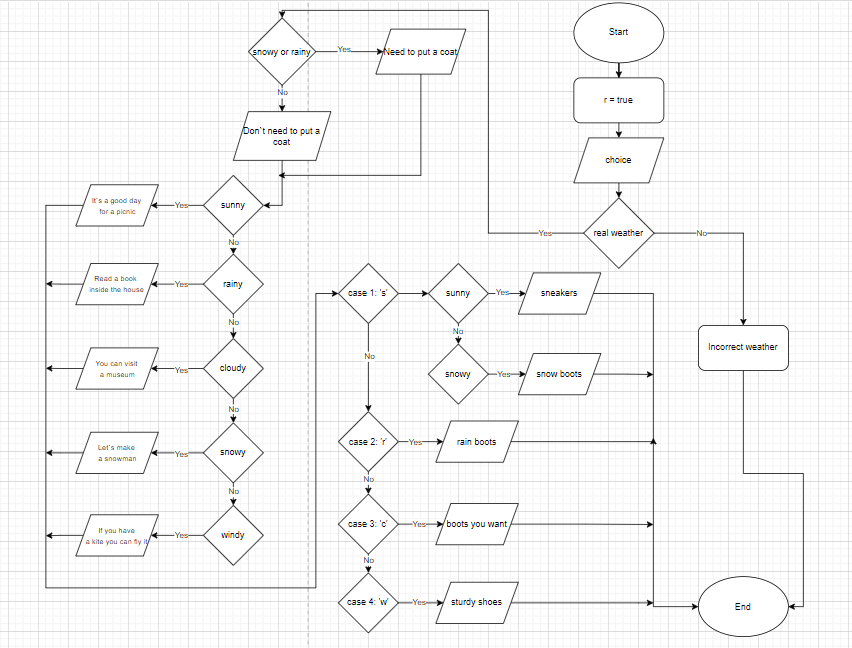
* Якщо сонячно, порекомендуйте «Чудовий день для пікніка!».
* Інакше, якщо буде дощ, рекомендуємо «Ідеальна погода, щоб читати книгу всередині!».
* Інакше, якщо хмарно, рекомендуємо «Може, відвідати музей?».
* Інакше, якщо сніг, порекомендуйте «Як щодо того, щоб зліпити сніговика?».
* Інакше, якщо буде вітер, порекомендуйте «Запустіть повітряного змія, якщо він у вас є!».

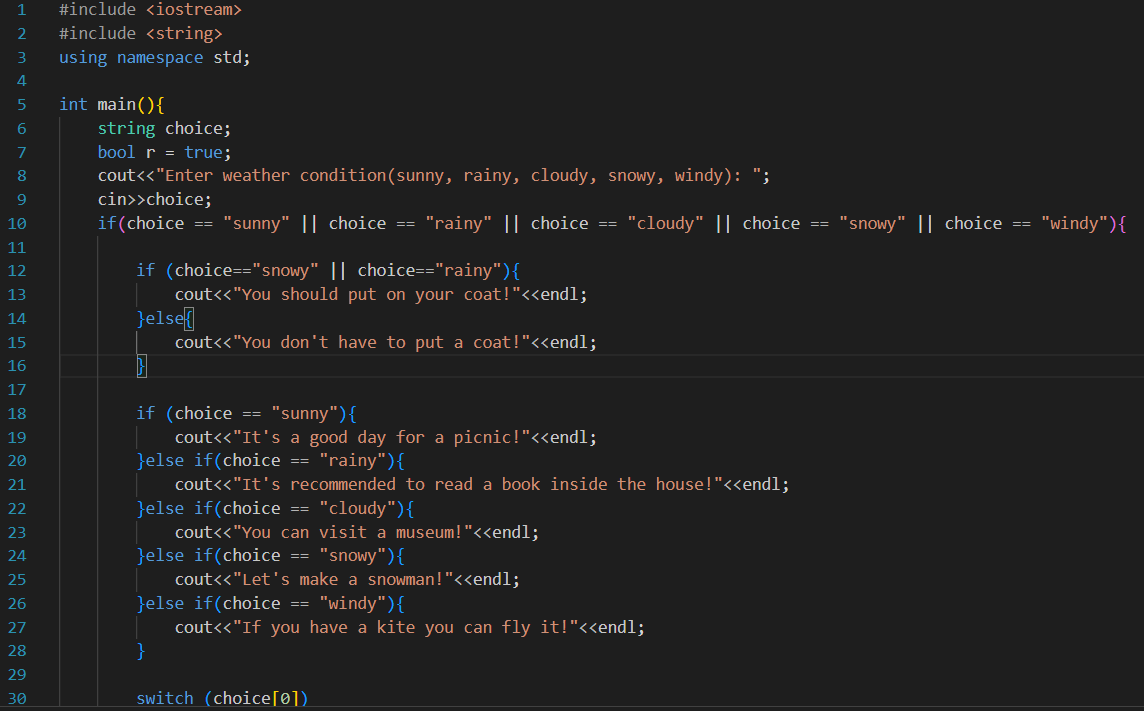
Рекомендації щодо взуття (з використанням футляра для вимикача)

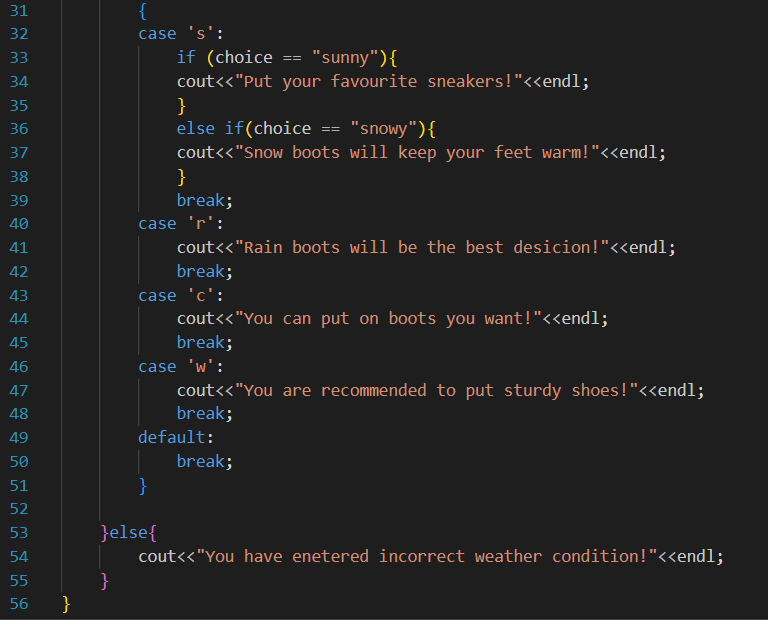
* sunny -> "Взуй улюблені кросівки!"
* rainy -> "Дощові чоботи - гарна ідея!"
* cloudy -> "Сьогодні підходить будь-яке взуття."
* snowy -> "Снігові черевики зігріють ваші ноги!"
* windy -> "Одягніть щось міцне!"

### Вимоги:

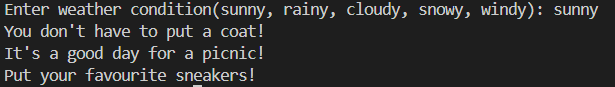
1. Використати всі згадані в передумові задачі оператори галуження - *if else, if, else if, switch case*;
2. За потреби комбінувати оператори;

**Блок-схема до задачі:**

**Розв’язок задачі:**

****

**Результат виконання програми:**

****

**Час виконання завдання ~ 25 хв**

**Завдання 4: Self Algotester Task 1**

У вашого персонажа є H хітпойнтів та M мани.

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

Якщо якесь 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.

# Вхідні дані

2 цілих числа

H та M - 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

# Вихідні дані

YES - якщо ваш персонаж виграє

NO - у всіх інших випадках

# Обмеження

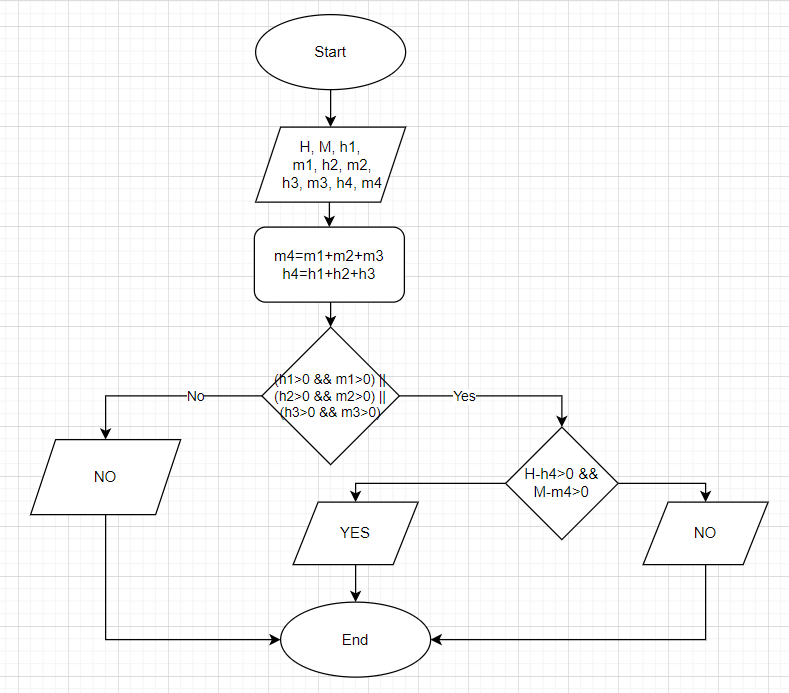
1≤H≤10^12

1≤M≤10^12

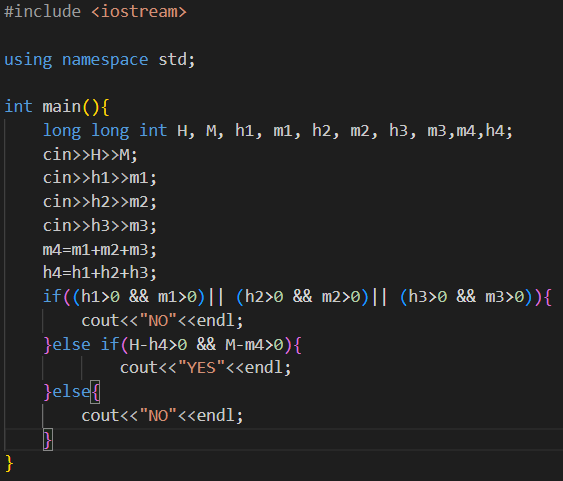
0≤hi≤10^12

0≤mi≤10^12

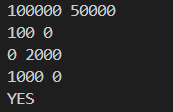
**Блок-схема до задачі:**

****

**Розв’язок задачі:**

****

**Результат виконання програми:**

****

**Час виконання завдання ~ 15 хв**

**Завдання 5: Self Algotester Task 2**

**Марічка і печиво**

Зібралися Зеник і Марічка разом з пластунами в похід. Похід — серйозна справа. Потрібно запастись продуктами харчування та розподілити їх споживання по днях так, щоб всім вистачило. Цього разу Зеник слідкує за тим, щоб печива вистачило аж до останнього дня походу. Зеник чітко знає, скільки пачок печива повинно залишитись кожного дня, і щовечора перераховує їх. Якщо Зеник побачить, що залишилось менше пачок, ніж повинно залишитись за його розрахунками, він неодмінно знайде того, хто з’їв забагато печива, і покарає його.

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.

**Вхідні дані**

У першому рядку задано одне натуральне число

n — кількість пачок печива.

У другому рядку задано n натуральних чисел ai— кількість штук печива в i-й пачці.

Вихідні дані

У єдиному рядку виведіть одне ціле число — максимальну кількість штук печива, яку зможе з’їсти Марічка так, щоб Зеник не помітив цього.

**Обмеження**

20% тестів:

1≤n≤1000,

0≤ai≤10^4

60% тестів:

1≤n≤10^5,

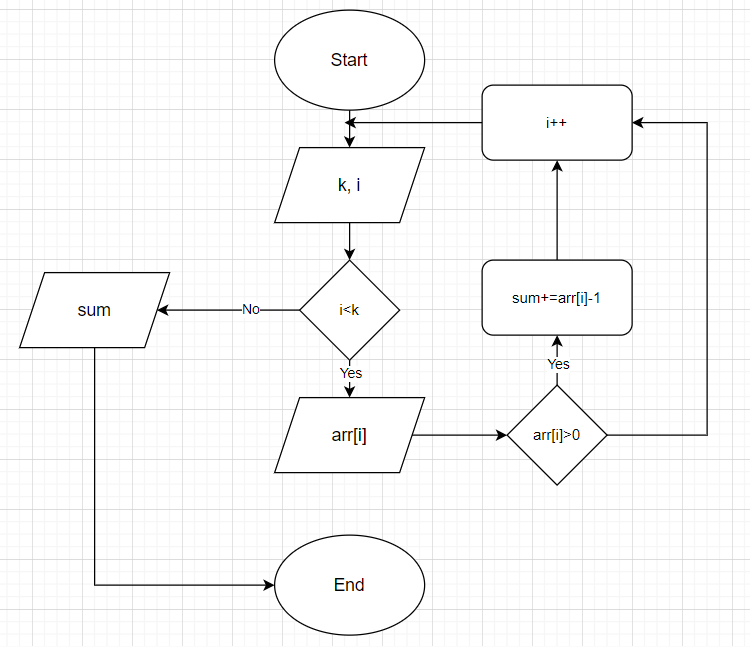
0≤ai≤10^4

20% тестів:

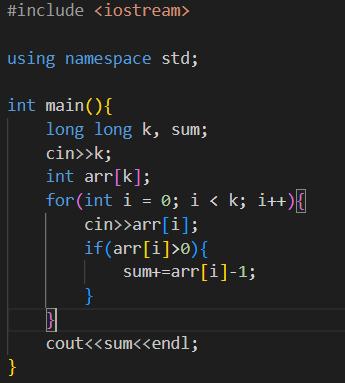
1≤n≤10^5,

0≤ai≤10^9

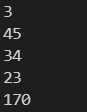
**Блок-схема до задачі:**

****

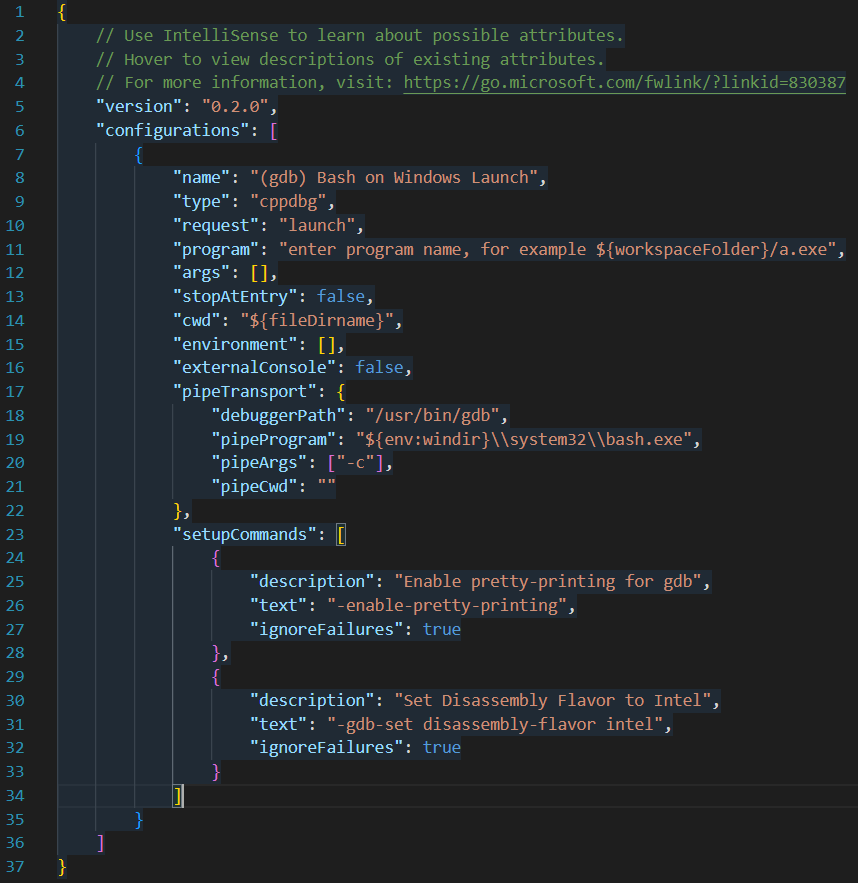
**Розв’язок задачі:**

****

**Результат виконання програми:**

****

**Час виконання завдання ~ 15 хв**

**Також налаштував файл launch.json**

**Висновок:**

В цьому епіку я навчився користуватися оператором if-else та switch. Навчився малювати перші блок-схеми до задач, розглянув цикли та зробив 2 додаткові задачі.