Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Звіт**

про виконання

**Лабораторних та практичних робіт № 5**

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

***з розділу***: «Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами»

***Виконала:***

студентка групи ШІ-13

Поліщук Марина Олександрівна

# **Тема роботи:**

" Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами"

# **Мета роботи:**

Навчитися працювати з файлами, системами числення, бінарними файлами, символами і рядковими змінними, текстовими файлами.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

-Тема №1: Ввід і вивід в С

-Тема №2: Стандартна бібліотека

-Тема №3: Системи числення

1. Індивідуальний план опрацювання теорії:

* Тема №1: Ввід і вивід в С
  + Джерела Інформації
    - лекції, практичні заняття
    - <https://t.me/c/1980904348/1/299?single>
  + Що опрацьовано:

У С відсутні засоби вводу-виводу. Всі операції вводу-виводу

реалізуються за допомогою функцій, які перебувають у бібліотеці С.

На рівні **потокового вводу-виводу** обмін даними відбувається побайтно,

тобто за одне звертання до пристрою (файлу) відбувається зчитування або

запис фіксованої порції даних (512 чи 1024 байти). При вводі з диска або при зчитуванні з файлу дані поміщаються в буфер ОС, потім побайтно або порціямимпередаються в програмі користувача. При виводі у файл дані накопичуються в буфері, а при заповненні буфера записуються у вигляді єдиного блоку на диск.

Буфери ОС реалізуються у вигляді ділянок основної пам'яті. Функції бібліотеки С, що підтримують обмін, з даними на рівні потоку дозволяють обробляти дані різних розмірів і форматів.

**Потік** - це файл разом з наданими засобами буферизації. При роботі з

потоком можна:

1) Відкривати й закривати потоки ( зв'язувати вказівники на потік з

конкретними файлами);

2) вводити і виводити рядок, символ, форматовані дані, порцію даних довільної довжини;

3) аналізувати помилки вводу-виводу й досягнення кінця файлу;

4) управляти буферизацією потоку й розміром буфера;

5) отримувати й встановлювати вказівник поточної позиції у файлі;

Функції бібліотеки вводу-виводу перебувають у заголовковому файлі <stdio.h>.

При відкритті потоку, повертається вказівник на потік, тобто на об'єкт типу FILE.

#include <stdio.h>;

. . . . . . . .

FILE \*fp;

. . . . . . . . . . ..

fp= fopen( ”t.txt”, ”r”;

де fopen(<ім'я\_файлу>,<режим\_відкриття>) - функція для ініціалізації файлу.

Існують такі режими для відкриття файлу:

”w” - відкрити файл для запису, якщо файл існує, то він стирається;

”r” - відкрити файл для читання;

”a” - відкрити файл для додавання, якщо файл існує, то він не стирається і

можна писати в кінець файлу;

”w+” - відкрити файл для запису й виправлення, якщо файл існує, то він

стирається, а далі можна і читати, і писати, розміри файлу можна збільшувати;

”r+” - відкрити файл для читання й запису, але збільшити розмір файлу не

можна;

”a+” - відкрити файл для додавання, тобто можна й читати й писати, у тому

числі й у кінець файлу.

Потік можна відкрити в текстовому (t) або двійковому (b) режимі. За

замовчуванням - текстовий режим. У явному виді режим вказується в такий

спосіб: ”r+b” або ”rb” - двійковий (бінарний) режим.

Для **блокового вводу і виводу** використовуються функції:

1) int fread( void \*ptr, int size, int n, FILE \*fp) , де

void \*ptr - вказівник на область пам'яті, у якій розташовуються зчитувані з

файлу дані;

int size - розмір одного зчитуваного елемента;

int n - кількість зчитуваних елементів;

FILE \*fp - вказівник на файл, з якого відбувається зчитування.

У випадку успішного зчитування інформації функція повертає число

прочитаних елементів (а не байтів), інакше повертає EOF.

2) int fwrite( void \*ptr, int size, int n, FILE \*fp) , де

void \*ptr - вказівник на область пам'яті, у якій розташовуються записувані у

файл дані;

int size - розмір одного записуваного елемента;

int n - кількість записуваних елементів;

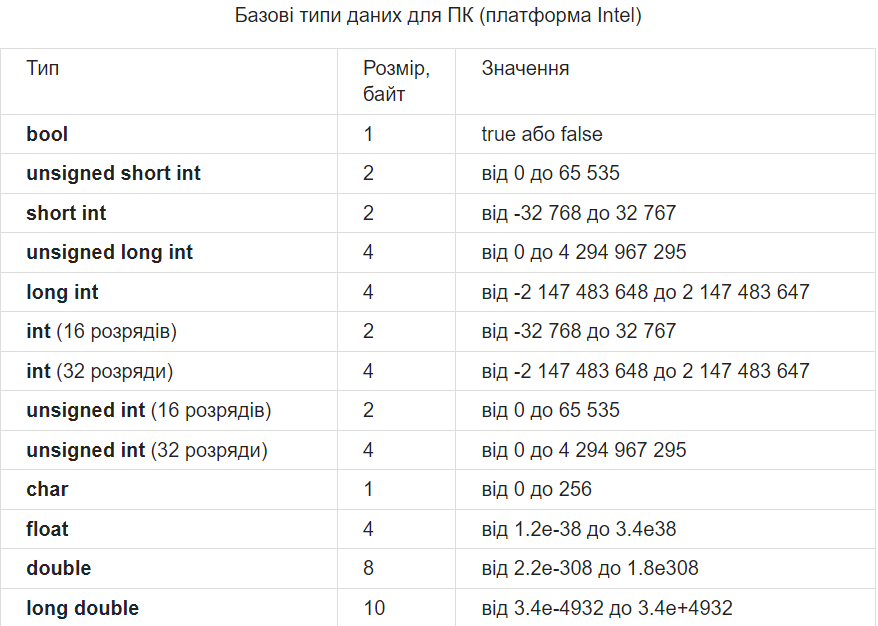
FILE \*fp - вказівник на файл, у який відбувається запис.

У випадку успішного запису інформації функція повертає число записаних

елементів, інакше повертає EOF.

Статус: Ознайомлена частково

* + - Початок опрацювання теми: 10.12.23
    - Звершення опрацювання теми: 13.12.23
* Тема №2: Типи змінних
  + Джерела Інформації
    - <http://cpp.dp.ua/typy-danyh/>
    - <https://t.me/c/1980904348/1/299?single>
  + Що опрацьовано:
    - Кожне з даних характеризується класом пам’яті, ім’ям, типом і значенням. Усі типи даних мови C++ можна розділити на основні (int (цілий), char (символьний), bool (логічний), float (дійсний), double (дійсний з подвійною точністю)) і складені (short (короткий), long (довгий), signed (знаковий), unsigned (беззнаковий)).



* + Статус: Ознайомлена
  + Початок опрацювання теми: 10.10.23
  + Звершення опрацювання теми: 13.10.23
* Тема №3: Системи числення
  + Джерела Інформації:
    - Відео

https://www.youtube.com/watch?v=c5FSYMpsLN0&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=67

* + Що опрацьовано:
    - 2-кова система числення
    - 8-кова система числення
    - 10-кова система числення
    - 16-кова система числення
  + Статус: Ознайомлений
  + Початок опрацювання теми: 11.12.2023
  + Звершення опрацювання теми: 11.12.2023
* Тема №4 Цикли for, while/do while
  + Джерела Інформації:
    - <https://drive.google.com/drive/u/0/folders/1rywFoh9PJBgf_vJn5EwSyo0Ywo7BvtI3>
    - лекційні, практичні заняття
  + Що опрацьовано:
    - цикл while (<вираз-умова>) <тіло\_циклу> -цикл з передумовою, якщо ж ми хочемо, щоб цикл виконався хоча б один раз можна використати цикл з післяумовою: do <тіло\_циклу>; while (<вираз-умова>).
    - цикл for (умова) {функція} виконується поки умова в () справджується
  + Статус: Ознайомлена
  + Початок опрацювання теми: 12.10.23
  + Звершення опрацювання теми: 17.10.23

# **Виконання роботи:**

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

Завдання №1 Опрацювання теорії

* Деталі завдання: опрацювання різноманітних матеріалів, аналіз та дослідження відео та статей на задані теми. Вивчення найважливіших моментів.

Завдання № 2 VNS lab 6 task 1

* Варіант завдання: 21
* Деталі завдання: Знищити всі парні слова у речені.
* Важливі деталі для врахування в імплементації програми: здійснено використовуючи функцію gets(s)

Завдання № 3 VNS lab 8 task 1

* Варіант завдання: 21
* Деталі завдання:

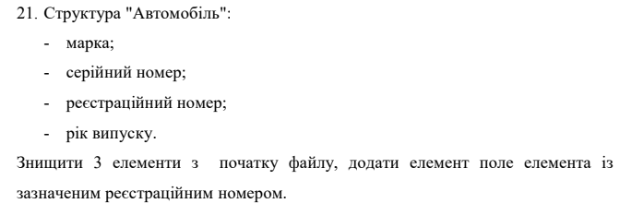


Рисунок 1 деталі завдання

* Важливі деталі для врахування в імплементації програми: знання як працюють постфіксні і префіксні операції. Підключено бібліотеки, оголошено змінні, обчислено і виведено вираз 1, тернарним оператором виведено чи правдиві нерівності 1 і 2

Завдання № 4 VNS lab 9 task 1

* Варіант завдання: 21
* Деталі завдання:

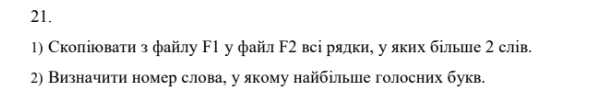


Рисунок 2 деталі завдання

* Важливі деталі для врахування в імплементації програми: знання як працюють постфіксні і префіксні операції. Підключено бібліотеки, оголошено змінні, обчислено і виведено вираз 1, тернарним оператором виведено чи правдиві нерівності 1 і 2

Завдання №5 Algotester Lab 4v2

* Варіант завдання: 2
* Деталі завдання:

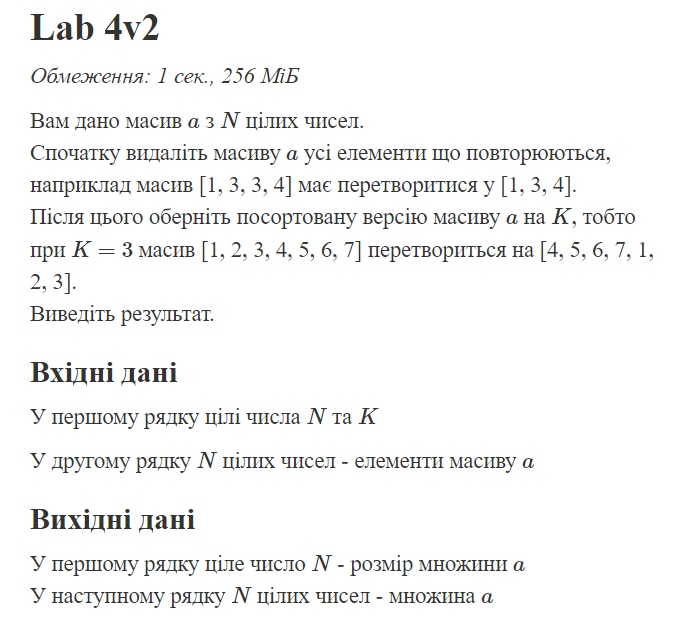


Рисунок 2 деталі завдання

* Важливі деталі для врахування в імплементації програми: написано код зі свою реалізацією без прописаних функцій.

Завдання №6 Algotester Lab 6v2

* Варіант завдання: 2
* Деталі завдання:

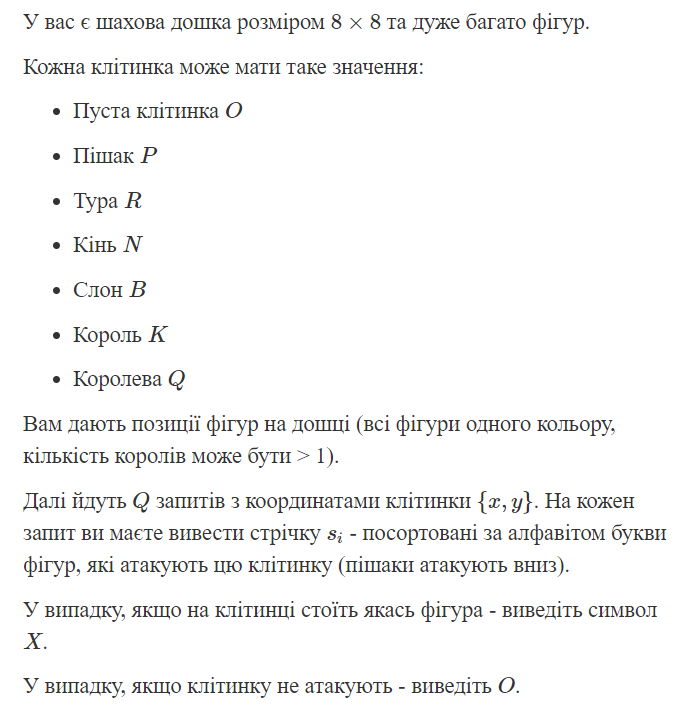


Рисунок 3 деталі завдання

* Важливі деталі для врахування в імплементації програми: використано багато того, чого я раніше не знала☺

Завдання №7 Class Practice Task 1

* Деталі завдання:

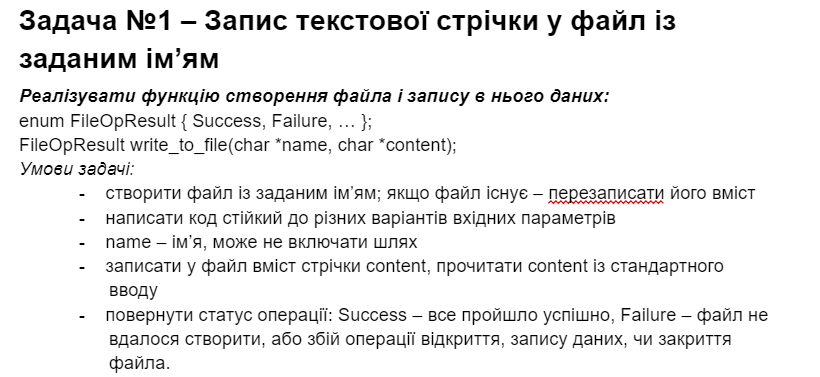


Рисунок 3 деталі завдання

Завдання №8 Class Practice Task 2

* Деталі завдання:

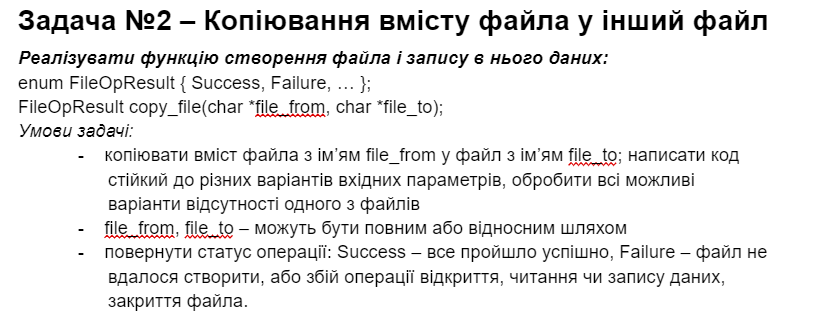


Рисунок 3 деталі завдання

Важливі деталі для врахування в імплементації програми:

Завдання №9 Self Practice Task

* Деталі завдання: Користувач вводить числа поки не введе нуль. Задача: обчислити суму всіх елементів, середнє арифметичне парних елементів. Підключено бібліотеки, створено змінні, користувачем вводиться число, додається до суми це число якщо воно парне, то до кількості парних елементів елементів додається один, до суми парних додається це число додається це число, якщо вводиться нуль, то від кількості елементів віднімається один (для коректного підрахунку елементів), обчислюється і виводить середнє арифметичне парних елементів.
* Важливі деталі для врахування в імплементації програми: використання циклу while (true), віднімати від кількості елементів 1 к оли введено нуль щоб був коректний підрахунок кількості елементів

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

Програма №1 VNS lab 1 task 1

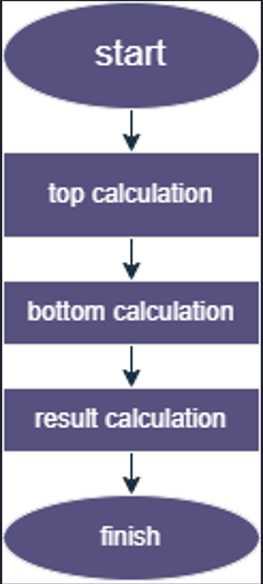
* Блок-схема
* 

Рисунок 4 VNS lab 1 task 1

* Планований час на реалізацію: 2 години
* Важливі деталі для врахування в імплементації: позначення проміжних етапів (чисельник та знаменник), різниця між використанням double i float.

Програма № 2 VNS lab 1 task 2

* Блок-схема

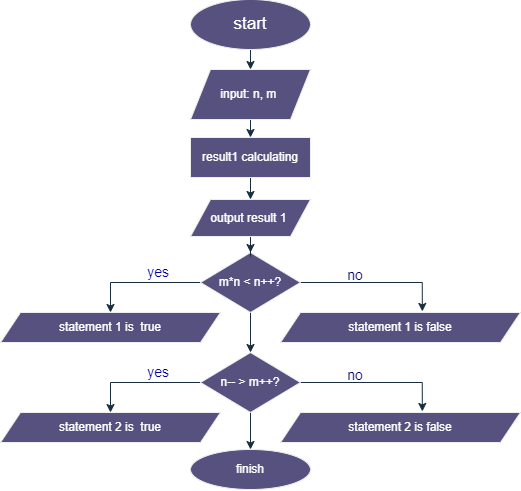


Рисунок 5 VNS lab 1 task 2

* Планований час на реалізацію: 3 години
* Важливі деталі для врахування в імплементації: знання як працюють постфіксні і префіксні операції.

Програма №3 Algotester Lab 1v3

* Блок-схема

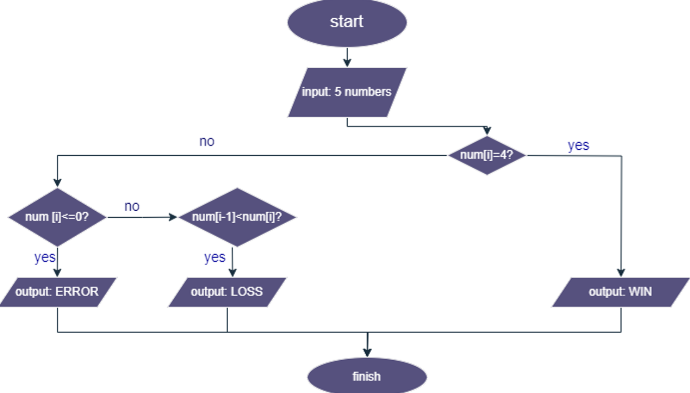


Рисунок 6 Algotester Lab 1v3

* Планований час на реалізацію: 6 годин
* Важливі деталі для врахування в імплементації: створення масиву для запису чисел, з його елементами перевіряються умови задачі.

Програма №4 Class Practice Task

* Блок-схема

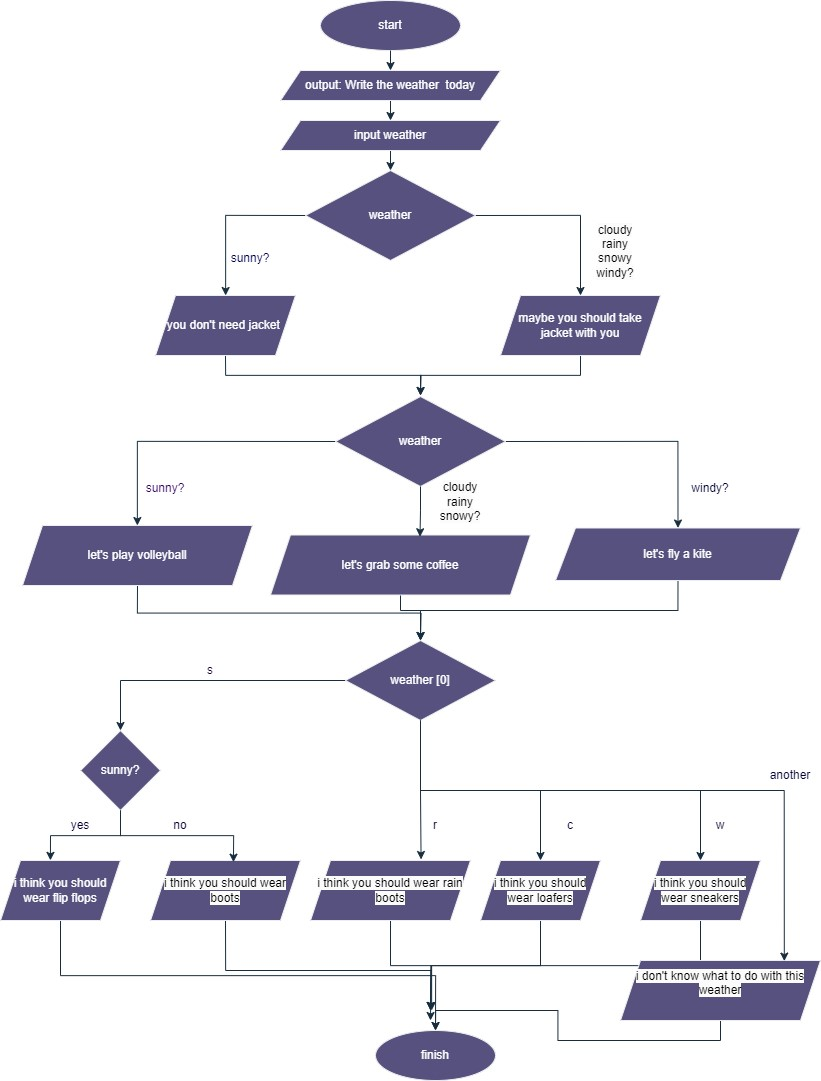


Рисунок 7 Class Practice Task

* Планований час на реалізацію: 4 години
* Важливі деталі для врахування в імплементації: використання switch case, if\ else if\ else

Програма №5 Self Practice Task

* Блок-схема

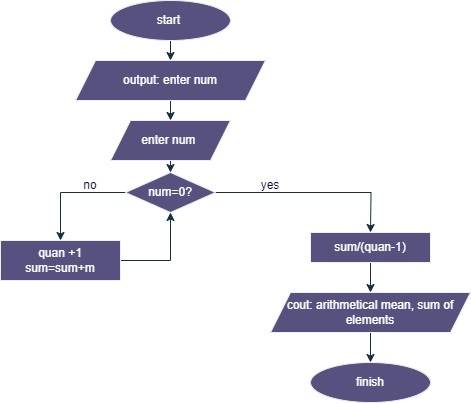


Рисунок 8 Self Practice Task

* Планований час на реалізацію: 4 години
* Важливі деталі для врахування в імплементації: використання циклу while (true)

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

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

Завдання № 1 vns lab1.1 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

//must be -1

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

double a, b, top, bottom, res;

a = 1000;

b = 0.0001;

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

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

res = top / bottom;

cout << res << endl;

return 0;

}

ai\_13/maryna\_polishchuk/epic2/lab1(2).cpp

## Завдання № 2 vns lab 1.2 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

## **int main ()**

## **{**

## **int m, n, result\_1;**

## **std::cout <<"enter m: ";**

## **std::cin >> m;**

## **std::cout <<"enter n: ";**

## **std::cin >> n;**

## **result\_1= --m-++n;**

## **std:: cout << "result\_1: " << result\_1 << std::endl;**

## **m\*n < n++ ? std::cout << "statement 1 is true\n" : std::cout << "statement 1 is false\n";**

## **n-- > m++ ? std::cout << "statement 2 is true\n" : std::cout << "statement 2 is false\n";**

## **return 0;**

## **}**

ai\_13/maryna\_polishchuk/epic2/lab 1.1.cpp

Завдання № 3 **algotester lab1- task 3** Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <iostream>

int main() {

long long num[5];

for(int i = 0; i < 5; i++)

{

std::cin >> num [i];

if (num [i]<=0)

{

std::cout<<"ERROR";

return 0;

}

else if (num[i-1]<num[i])

{

std:: cout << "LOSS";

return 0;

}

}

std:: cout <<"WIN";

return 0;

}

## **ai\_13/maryna\_polishchuk/epic2/algotester lab1- task 3.cpp**

## Завдання № 4 class practice

## Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <string>

using namespace std;

int main (){

string weather;

cout << "Write the weather today: " << endl;

cin >> weather;

if ( weather=="sunny")

{

cout << "You don't need jacket" << endl;}

else if (weather=="cloudy"||weather=="rainy"||weather=="snowy"||weather=="windy")

{

cout << "Maybe you should take jacket with you" << endl;

}

if (weather=="sunny")

cout << "Let's play volleyball" << endl;

else if (weather=="cloudy"||weather=="rainy"||weather=="snowy")

{

cout << "Let's grab some coffee"<<endl;

}

else if (weather=="windy")

{

cout << "Let's fly a kite" << endl;

}

switch (weather[0]) {

case 's':

if (weather=="sunny")

cout << "I think you should wear flip flops" << endl;

else

cout << "I think you should wear boots"<< endl;

break;

case 'r':

cout << "I think you should wear rain boots" << endl;

break;

case 'c':

cout << "I think you should wear loafers" << endl;

break;

case 'w':

cout << "I think you should wear sneakers" << endl;

break;

default:

cout << "I don't know what to do with this weather" << endl;

break;

}

return 0;

}

ai\_13/maryna\_polishchuk/epic2/pr\_epic\_2.cpp

Завдання № 5 self practice Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

//Користувач вводить числа поки не введе нуль.

//Задача: обчислити суму всіх елементів, середнє арифметичне парних елементів.

#include <iostream>

int main()

{

int sum=0; // сума усіх елементів

int evenSum=0; //сума арних елементів

double quan=0; //кількість парних елементів

double ArifMean; //середнє арифметичне парних елементів

while (true)

{

int m;

std::cout <<"enter num: ";

std::cin >> m;

if(m%2==0)

{

evenSum+=m;

quan++;

}

sum+=m;

if (m==0)

{

quan--;

ArifMean= evenSum/quan;

std:: cout << "arithmetical mean: " <<ArifMean << std::endl;

std:: cout << "Sum of elements: " <<sum;

return 0;

}

}

return 0;

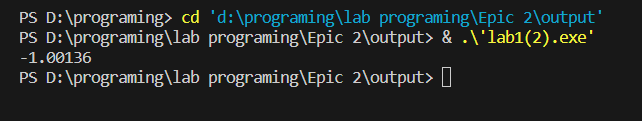
}

ai\_13/maryna\_polishchuk/epic2/sum2.cpp

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

Завдання № 1 VNS lab 1 task 1

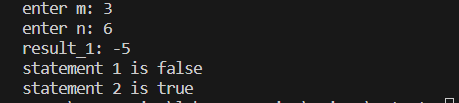
Деталі по виконанню і тестуванню програми



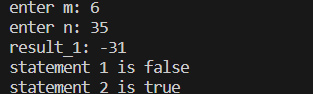
Час затрачений на виконання завдання: 6 годин

Завдання № 2 VNS lab 1 task 2

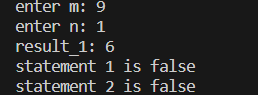
Деталі по виконанню і тестуванню програми



тест 1

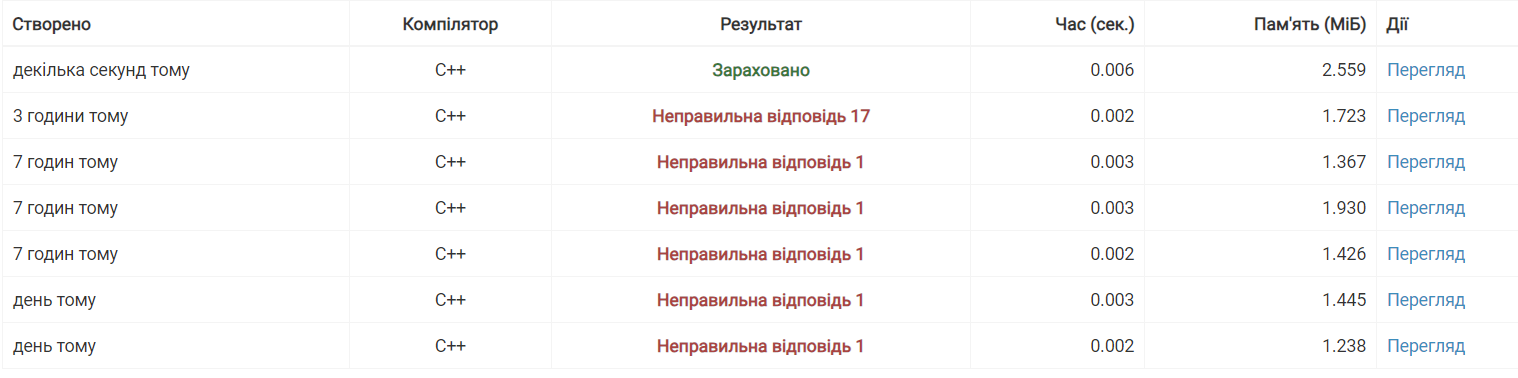


тест 2



Час затрачений на виконання завдання: 1.5 години

Завдання № 3 Algotester Lab 1v3

Деталі по виконанню і тестуванню програми 



тест 1



тест 2

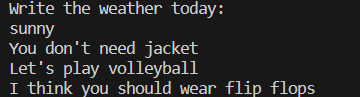


тест 3

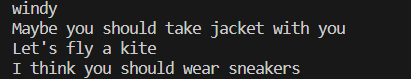
Час затрачений на виконання завдання: 7 годин

Завдання № 4 Class Practice Task

Деталі по виконанню і тестуванню програми



тест 1



тест 2



тест 3

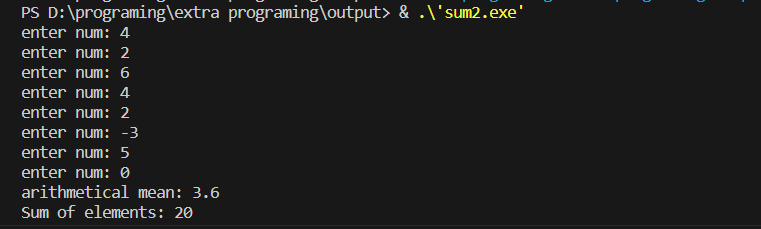
Час затрачений на виконання завдання: 6 годин

Завдання №5 Self Practice Task

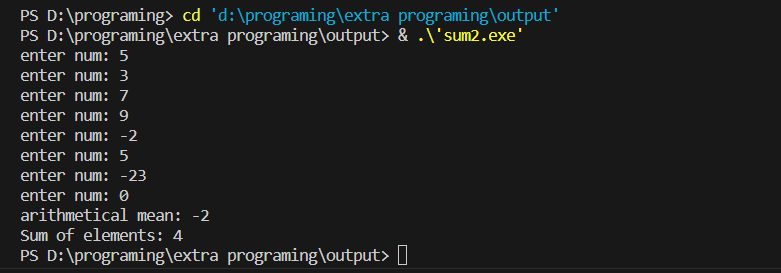
Деталі по виконанню і тестуванню програми



тест 1



тест 2



тест 3

Час затрачений на виконання завдання: 5 годин

# **Висновки:**

Ми навчилися працювати з лінійними та розгалуженими алгоритмами та умовними операторами, а також з константами та змінними.