Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №2**

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

***з розділу***: “Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні”

***Виконав:***

студент групи ШІ-14

Шелеп Андрій Іванович

**Тема роботи:**

Знайомство з С/С++. Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні

**Мета роботи:**

Ознайомитися з лінійними та розгалуженими алгоритмами, типами даних, змінними та константами, конструкціями умови та умовними операторами та застосувати їх.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Типи даних, змінні, константи
* Тема №2: Умовні оператори

1. Індивідуальний план опрацювання теорії:

* Тема №1: Типи даних, змінні, константи.
  + Джерела Інформації
    - <http://cpp.dp.ua/typy-danyh/>
    - <https://purecodecpp.com/uk/archives/165>
    - <https://www.w3schools.com/cpp/cpp_function_overloading.asp>
  + Що опрацьовано:
    - типи даних у С++, їх розміри і значення. Константи і змінні, їх задання
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.10.2023
  + Завершення опрацювання теми: 30.10.2023
* Тема №2: Умовні оператори.
  + Джерела Інформації:
    - <https://lib.chmnu.edu.ua/pdf/posibnuku/350/6.pdf>
    - <https://www.redcourses.pp.ua/%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%B8-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F/c>
    - <https://www.youtube.com/watch?v=zogwWqGyM2c&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
    - <https://www.youtube.com/watch?v=OuPiJ5SMYis&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
  + Що опрацьовано:
    - Оператори розгалуження (if, if else, else if, switch) в С++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.10.2023
  + Завершення опрацювання теми: 30.10.2023

**Виконання роботи:**

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

Завдання №1 VNS Lab 1 Task 1

* Варіант 12
* Деталі завдання: Обчислити значення виразу при різних дійсних типах даних (float й double)

, при а=1000, b=0.0001

* потрібно пам’ятати про розмір і точність типу даних float
* Важливі деталі для врахування в імплементації: Для вводу й виводу даних використати операції >> й << і стандартні потоки cin й cout. Використовувати pow(). Використовувати проміжні змінні.

Завдання №2 VNS Lab 1 Task 2

* Варіант 12
* Деталі завдання: 1) n+++m 2) m-- >n 3) n-- >m
* Важливі деталі для врахування в імплементації: знати як працюють пре(пост) інкременти і дикременти

Завдання №3 Algotester Lab 1 Task 2

* Варіант 2
* Деталі завдання: Є стіл з 4 ніжками від яких відпилюють відповідні довжини. Потрібно написати програму, яка скаже, чи стіл буде стояти паралельно підлозі

Завдання №4 Class Practice Task

* Деталі завдання: Порадник, який залежить від погоди. В залежності від того, який тип погоди ввів користувач, програма дає пораду щодо одягу, взуття та активності.
* Важливі деталі для врахування в імплементації: для того, щоб порадити чи брати куртку, потрібно використовувати if else, щоб порадити активність - else if, щоб порадити взуття - switch case.

Завдання №5 programming: Self Practice Task

* Задача з алготестеру 031 “Коля, Вася і тенія”
* Деталі завдання: Всього відбулося n подач. Про кожну з них ми знаємо, хто переміг. За виграну подачу гравець отримує одне очко. Партія вважається виграною, коли один з гравців набере не менше одинадцяти очок з перевагою щонайменше у два очки. Наприклад, за рахунків 11:9, 4:11, 15:13 партія закінчується, а за рахунків 11:10 та 99:98 — ні. Як тільки Коля і Вася закінчили одну партію — вони починають іншу.

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

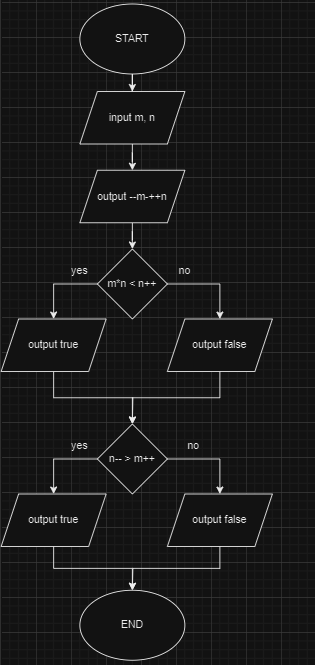
Програма №1 №1 VNS Lab 1 Task 1

блоксхема


*Image* *1. блок-схема для програми VNS Lab 1 Task 1*

* Планований час на реалізацію : 15 хв

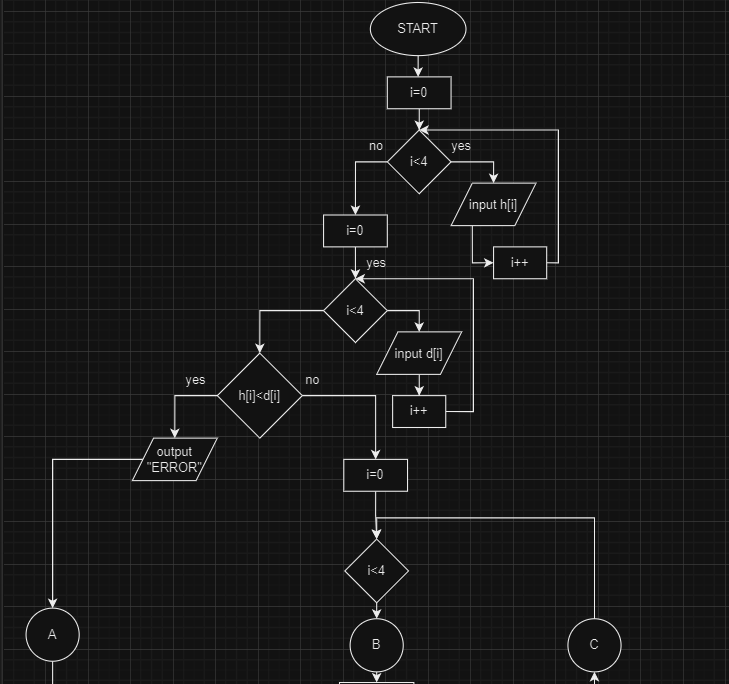
Програма №2 VNS Lab 1 Task 2

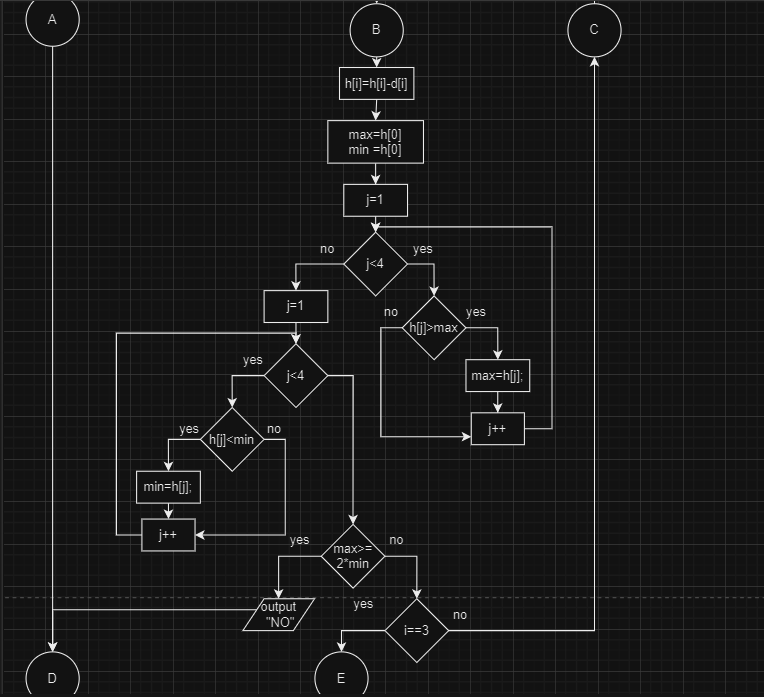


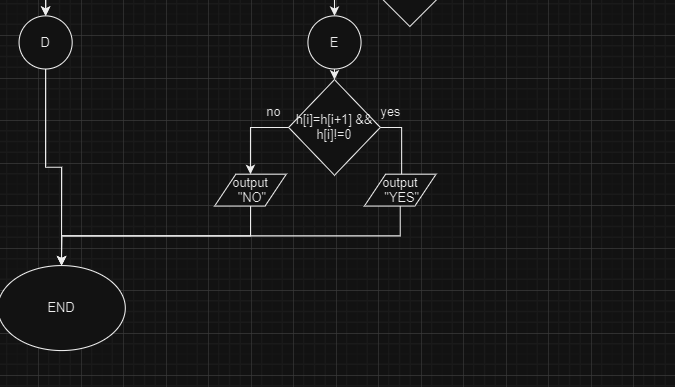
*Image* *2. блок-схема для програми VNS Lab 1 Task1*

* Планований час на реалізацію : 15 хв

Програма №3 Algotester Lab 1 Task 2



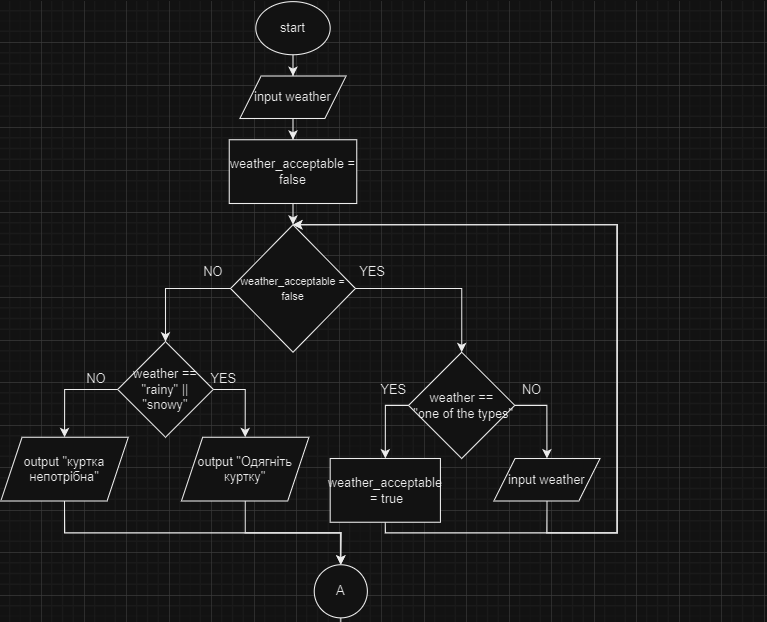


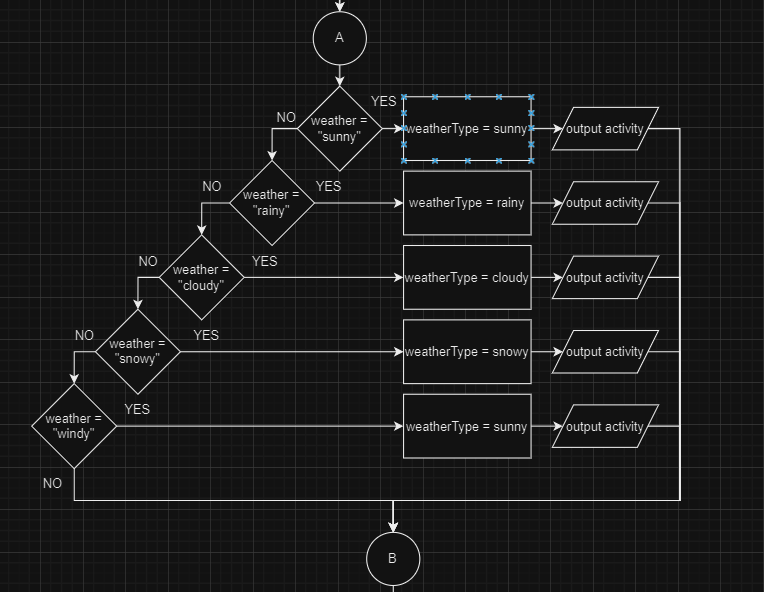


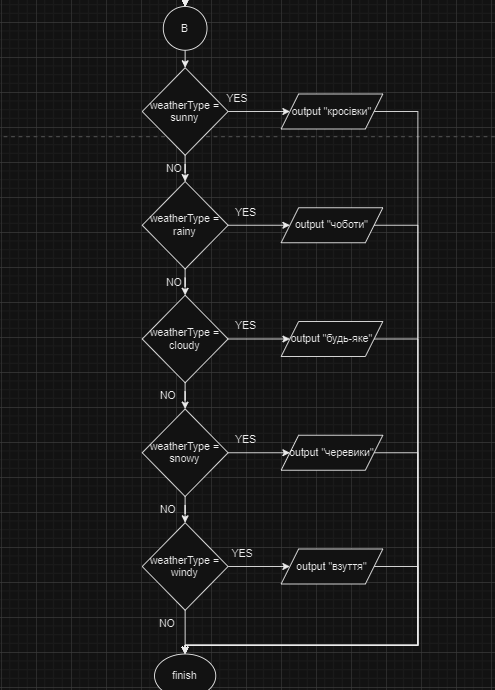
*Image* *3. блок-схема для програми algotester Lab 1*

* Планований час на реалізацію : 2 год

Завдання №4 programming: Class Practice Task



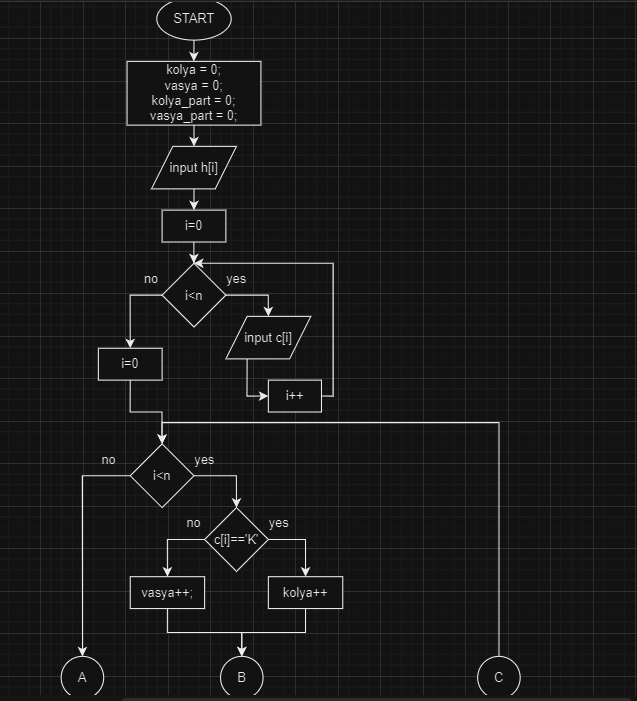


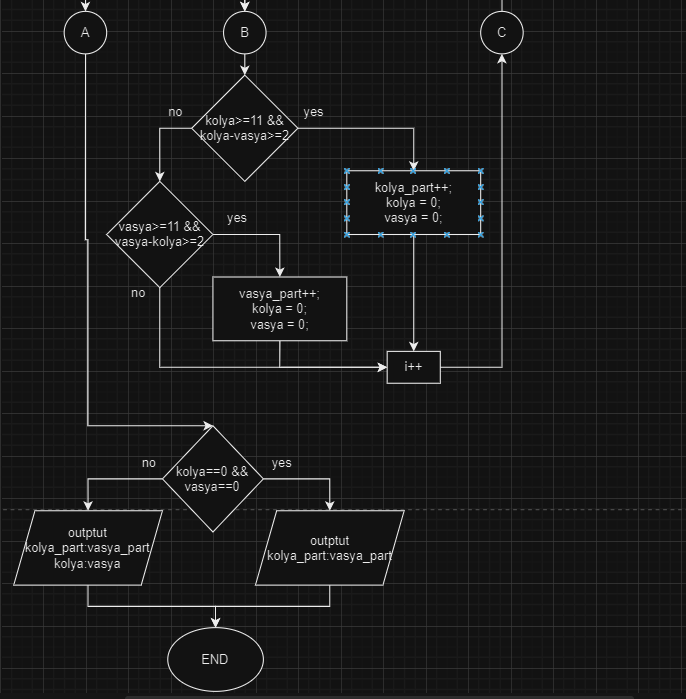


*Image* *4. блок-схема для програми class practice task*

* Планований час на реалізацію : 1 пара

Завдання №5 programming: Self Practice Task





*Image* *5. блок-схема для програми class practice task*

Рисунок 5. Блоксхема до програми №5 self practice work

* Планований час на реалізацію : 45 хв

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

Використано налаштування з минулої лабораторної.

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

посилання на pull request <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/219>

Завдання №1 VNS Lab 1 Task 1

#include <iostream>

#include <math.h>

using *namespace* std;

*int* main(){

*double* a=1000, b=0.0001, c, d, e, f;

c = pow(a+b,2);

d = pow(a,2);

e=2\*a\*b;

f=pow(b,2);

*double* r = (c-(d+e))/f;

cout << “double:” << r << endl;

return 0;

}

#include <iostream>

#include <math.h>

using *namespace* std;

*int* main(){

*float* a=1000, b=0.0001, c, d, e, f;

c = pow(a+b,2);

d = pow(a,2);

e=2\*a\*b;

f=pow(b,2);

*float* r = (c-(d+e))/f;

cout << "float:" << r << endl;

return 0;

}

Завдання №2 VNS Lab 1 Task 2

#include <iostream>

using *namespace* std;

*int* main()

{

*int* n, m;

cout << "Enter m, n\n";

cin >> m >> n;

cout << "1) --m-++n = " << --m-++n << endl;

//тепер m=m-1; n=n+1

if (m\*n < n++){

cout << "2) m\*n < n++: " << true << endl;

}

else{

cout << "2) m\*n < n++: " << false << endl;

}

//тепер n=n+1

if (n-- > m++){

cout << "3) n-- > m++: " <<true << endl;

}

else{

cout << "3) n-- > m++: " <<false << endl;

}

//тепер n=n-1; m=m+1

return 0;

}

Завдання №3 Algotester Lab 1 Task 2

#include <iostream>

using *namespace* std;

*int* main(){

*unsigned* *long* *long* h[4], d[4], max, min;

*int* i;

// вводимо довжини ніжок

for (i = 0; i < 4; i++){

cin >> h[i];

}

// вводимо довжини,які будемо відпилювати

for (i = 0; i < 4; i++){

cin >> d[i];

}

// перевіряємо чи довжина відпилювання не більша за відповідну ніжку

if (h[0]<d[0] || h[1]<d[1] || h[2]<d[2] || h[3]<d[3]){

cout << "ERROR";

} else{

for (i = 0; i < 4; i++){

// по черзі відпилюємо від ніжок відповідні довжини

h[i]-=d[i];

max = h[0];

min = h[0];

// після кожного відпилювання шукаємо максимальну і мінімальну довжину ніжок

for (*int* j = 1; j < 4; j++){

if (h[j] > max){

max = h[j];

}

}

for (*int* j = 1; j < 4; j++){

if (h[j] < min){

min = h[j];

}

}

// та перевіряємо чи максимальна довжина є вдвічі більшою рівна за мінімальну

if (max >= 2\*min){

cout << "NO" << endl;

break;

} else if (i==3){

// якщо це остання ітерація циклу, перевіряємо чи всі ніжки рівні після вдалих відпилювань (якщо так - стіл буде паралельний підлозі)

if (h[0]==h[1] && h[1]==h[2] && h[2]==h[3] && h[0] != 0 && h[1] != 0 && h[2] != 0 && h[3] != 0){

cout << "YES";

} else {

cout << "NO";

}

}

}

}

return 0;

}

Завдання №4 programming: Class Practice Task

#include <iostream>

#include <string>

using *namespace* std;

*enum* *WeatherTypes* {sunny, rainy, cloudy, snowy, windy};

*int* main(){

*string* weather;

*int* weatherType;

cout << "enter the type of weather (sunny, rainy, cloudy, snowy, windy):" << endl;

cin >> weather;

//перевіряємо чи користувач ввів правильний тип погоди

*bool* weather\_acceptable = false;

while(weather\_acceptable==false){

if(weather == "sunny" || weather == "rainy" || weather == "cloudy" || weather == "snowy" || weather == "windy"){

weather\_acceptable = true;

}

// якщо ні - просимо, що він ще раз ввів його

else{

cout << "введіть правильне значення: ";

cin >> weather;

}

}

// в залежності від погоди, програма радить чи надягати куртку

if (weather == "rainy" || weather == "snowy"){

cout << "Одягніть куртку!" << endl;

} else{

cout << "Куртка непотрібна!" << endl;

}

// в залежності від погоди, програма радить якусь активність

if (weather == "sunny"){

weatherType = *WeatherTypes*::sunny;

cout << "Чудовий день для пікніка!" << endl;

} else if (weather == "rainy"){

weatherType = *WeatherTypes*::rainy;

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

} else if (weather == "cloudy"){

weatherType = *WeatherTypes*::cloudy;

cout << "Може, відвідати музей?" << endl;

} else if(weather == "snowy"){

weatherType = *WeatherTypes*::snowy;

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

} else if (weather == "windy"){

weatherType = *WeatherTypes*::windy;

cout << "Запустіть повітряного змія, якщо він у вас є!" << endl;

}

// в залежності від погоди, програма радить кросівки

switch (weatherType)

{

case *WeatherTypes*::sunny:

cout << "Взуй улюблені кросівки!";

break;

case *WeatherTypes*::rainy:

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

break;

case *WeatherTypes*::cloudy:

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

break;

case *WeatherTypes*::snowy:

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

break;

case *WeatherTypes*::windy:

cout << "Взуйте щось міцне!";

break;

}

}

Завдання №5 programming: Self Practice Task

//031 Коля, Вася і теніс

#include <iostream>

using *namespace* std;

*int* main(){

*char* c[100000];

*int* n, kolya=0, vasya=0, kolya\_part=0, vasya\_part=0;

cin >> n; //вводимо к-ть подач

for (*int* i=0; i<n; i++){

cin >> c[i]; //вводимо подачі

}

for (*int* i=0; i<n; i++){

if(c[i]=='K'){ //якщо подачу виграв Коля, то добавляємо йому очко до рахунку

kolya++;

} else{ //якщо подачу виграв Вася, то добавляємо йому очко до рахунку

vasya++;

}

if (kolya>=11 && kolya-vasya >= 2){ //якщо Коля набрав 11 чи більше, і рахунок відрізняється мінімум на 2 очки, то Коля виграв партію і рахунок подач онуляється

kolya\_part++;

kolya=0;

vasya=0;

} else if(vasya>=11 && vasya-kolya >= 2){ //якщо Вася набрав 11 чи більше, і рахунок відрізняється мінімум на 2 очки, то Вася виграв партію і рахунок подач онуляється

vasya\_part++;

kolya=0;

vasya=0;

}

}

// тоді виводимо рахунок, якщо всі партії дограні, то виводимо тільки рахунок по партіях, якщо ні - то ще рахунок по подачах

if (kolya==0 && vasya==0){

cout << kolya\_part << ":" << vasya\_part << endl;

} else{

cout << kolya\_part << ":" << vasya\_part << endl << kolya << ":" << vasya << endl;

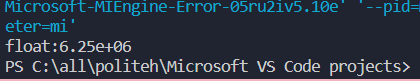
}

return 0;

}

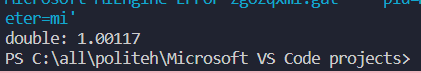
**5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 VNS Lab 1 Task 1



*Image 6. Результати виконання програми №1*

Виводить таке число через розмір розмір типу даних float та його точністю. Піднесення до квадрата скоріш за все призводить до втрати точності через обмежену кількість бітів у представленні чисел типу float

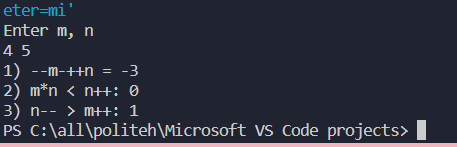


*Image 7. Результати виконання програми №1*

double має більші розмір і точність, тому виводить точніше значення

Час затрачений на виконання завдання: 20 хв

Завдання №2 VNS Lab 1 Task 2



*Image 8. Результати виконання програми №2*

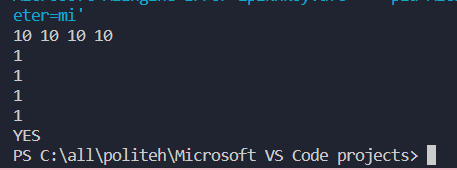
на першому етапі m=m-1=3, n=n+1=6 -> 3-6=-3

на другому етапі 3\*6=18 18>6 -> false

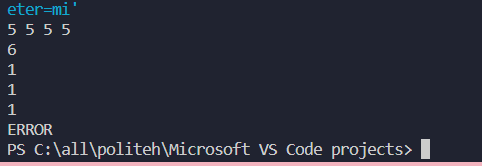
тоді n=n+1=7. На третьому етапі 7>3 -> true

Час затрачений на виконання завдання: 20 хв

Завдання №3 Algotester Lab 1

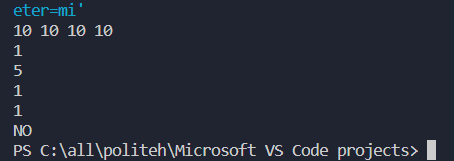


*Image 9. Результати виконання програми №3*



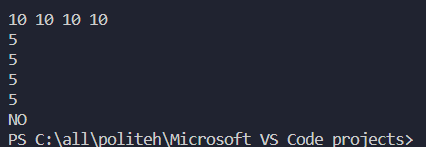
*Image 10. Результати виконання програми №3*

у цьому випадку довжина відпилювання більша за довжину ніжки



*Image 11. Результати виконання програми №3*

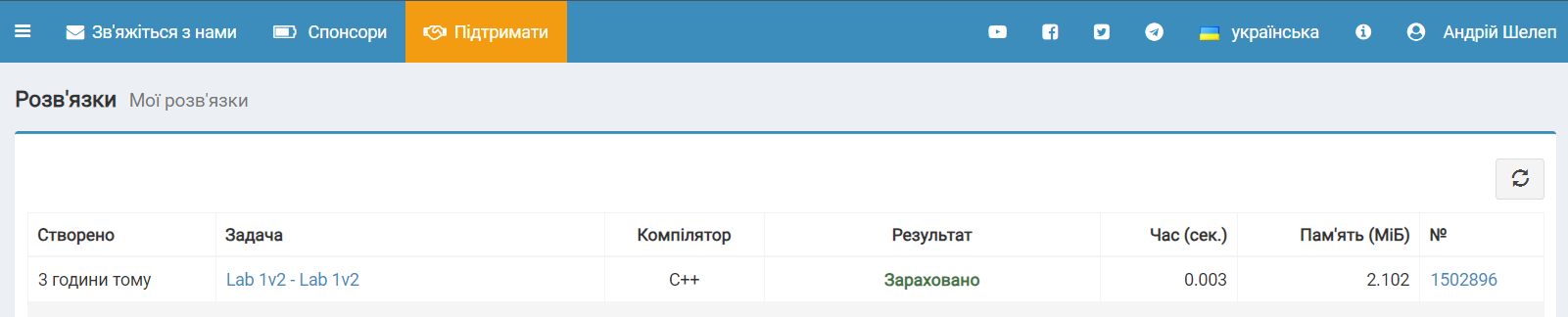
у цьому випадку після 2-го відпилювання друга ніжка буде вдвічі меншою за максимальну



*Image 12. Результати виконання програми №3*

у цьому випадку після 1-го відпилювання друга ніжка буде вдвічі меншою за максимальну

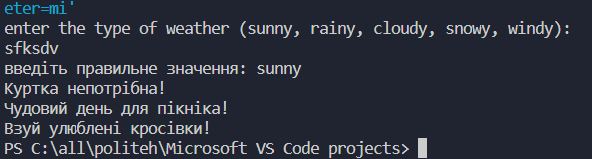
Час затрачений на виконання завдання: 2-2,5 год



*Image 13. Результати виконання програми №3*

Прийнято на алготестері

Завдання №4 Class-practice work

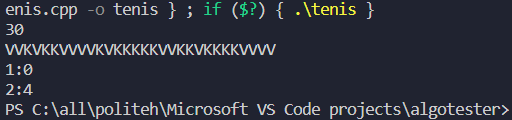


*Image 14. Результати виконання програми №4.*

Спочатку ввели неправилбний тип погоди, тому програма попросила ввести ще раз. І відповідно до типу погоди sunny програма видала пораду

Час затрачений на виконання завдання 1 пара

Завдання №7 Self-practice work



*Image 15. Результати виконання програми №5*

Спочатку Коля виграв з рахунком 13:11, тоді наступну партію недограли з рахунком 2:4 на користь Васі

Час затрачений на виконання завдання 1 год.

**Висновки:**

Розв’язано задачі з ВНС та алготестеру, написано код програми, яку робили на практичній. Ознайомився з розгалуженими алгоритмами, типами даних, операторами умови, циклу попрацював з ними, розв’язуючи задачі