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 типами даних; ознайомитися з циклами та умовними операторами та застосувати їх.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Типи даних
* Тема №2: Оператори умови
* Тема №3: Цикли
* Тема №4: Сума ряду із заданою точністю.
* Тема №5: Сортування бульбашкою

1. Індивідуальний план опрацювання теорії:

* Тема №1: Типи даних.
  + Джерела Інформації
    - <http://cpp.dp.ua/typy-danyh/>
  + Що опрацьовано:
    - типи даних у С++, їх розміри і значення
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.10.2023
  + Завершення опрацювання теми: 30.10.2023
* Тема №2: Оператори умови.
  + 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>
  + Що опрацьовано:
    - Оператори умови в С++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.10.2023
  + Завершення опрацювання теми: 30.10.2023
* Тема №3: Цикли.
  + Джерела Інформації:
    - <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>
    - <http://cpp.dp.ua/operatory-tsyklu/>
    - <https://www.youtube.com/watch?v=rj1OLsBKazA&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=UY295pIdeoQ&t=437s&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=31NvnS_5n1w&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:
    - Цикли (while, do while, for). Оператори break, continue, goto
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.10.2023
  + Завершення опрацювання теми: 30.10.2023
* Тема №4 Сума ряду із заданою точністю.
  + Джерела Інформації:
    - <https://www.youtube.com/watch?v=6Es6pmKBdTY&ab_channel=IrinaBlazhievska>
    - <https://ela.kpi.ua/bitstream/123456789/41853/1/VyshchMat_Chyslovi-ta-funktsionalni-riady_Praktykum.pdf>
  + Що опрацьовано:
    - Збіжні ряди та їх сума із заданою точністю
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.10.2023
  + Завершення опрацювання теми: 30.10.2023
* Тема №4 Сортування бульбашкою.
  + Джерела Інформації:
    - <https://www.programiz.com/dsa/bubble-sort>
  + Що опрацьовано:
    - Один з способів сортування масиву
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.10.2023
  + Завершення опрацювання теми: 20.10.2023

# **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)

(-(2ab+))/2, при а=1000, b=0.0001

* потрібно пам’ятати про розмір і точність типу даних float

Завдання №2 VNS Lab 1 Task 2

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

Завдання №3 VNS Lab 2 Task 1

* Варіант 14
* Деталі завдання: обчислити суму ряду an=n!/3\*n^n з точністю 0.0001

Завдання №4 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 №5 Algotester Lab 2 Task 1

* Варіант 1
* Деталі завдання: Є дорога (масив) з N чисел, і ми можемо порахувати втому, віднявши від максимального числа мінімальне. Написати програму, яка виводить мінімальну втому при відкидуванні одного числа з дороги(масиву).

Завдання №6 programming: Class Practice Task

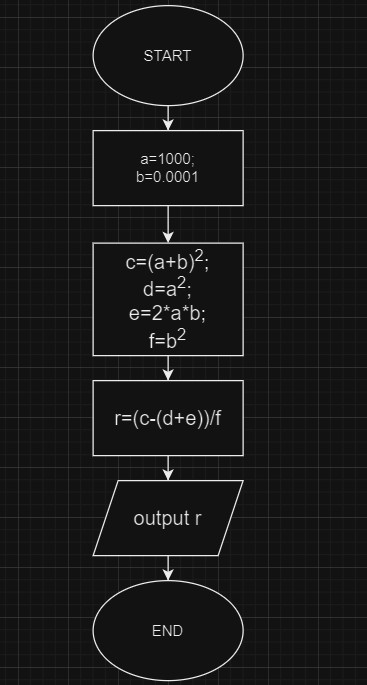
* Деталі завдання: Порадник, який залежить від погоди. В залежності від того, який тип погоди ввів користувач, програма дає пораду щодо одягу, взуття та активності.

Завдання №7 programming: Self Practice Task

* Задача з алготестеру “Веселі каруселі” 0531
* Деталі завдання: Відомо, що Марічка має k гривень, а в парку є n атракціонів. Для кожного атракціону відома його ціна c грн. Вам необхідно визначити кількість атракціонів, на яких покатається Марічка (Марічка йде на атркціони від дешевшого до дорожчого).

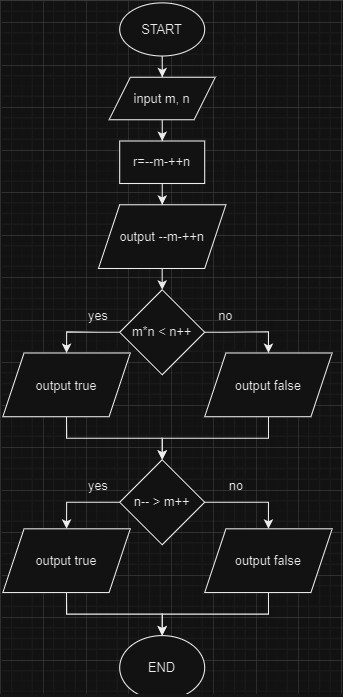
## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №1 №1 VNS Lab 1 Task 1



* Планований час на реалізацію : 15 хв

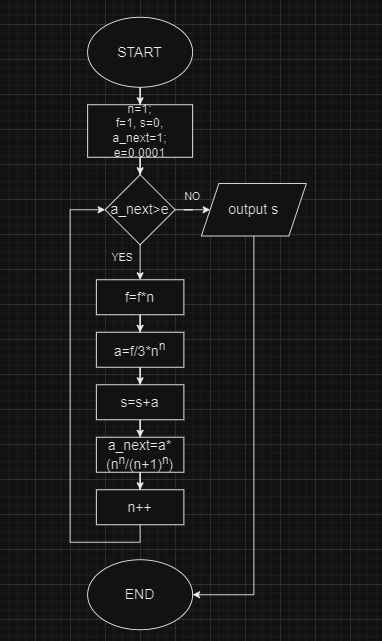
Програма №2 VNS Lab 1 Task 2



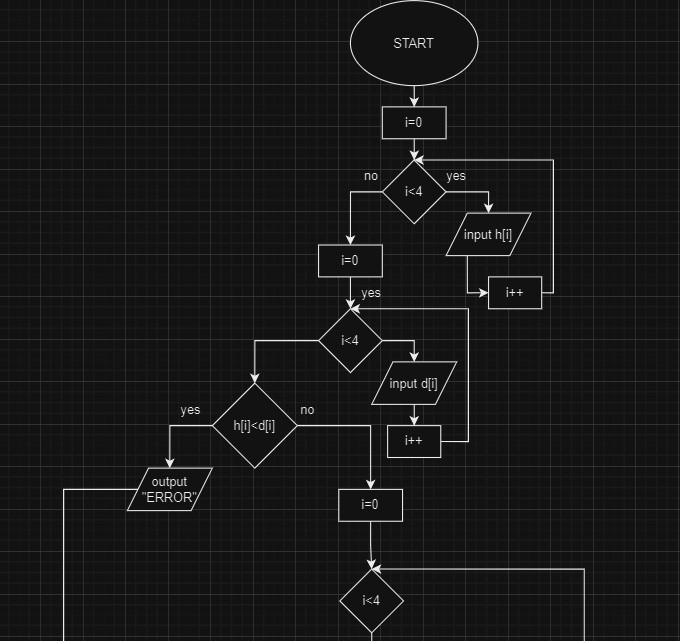
* Планований час на реалізацію : 15 хв

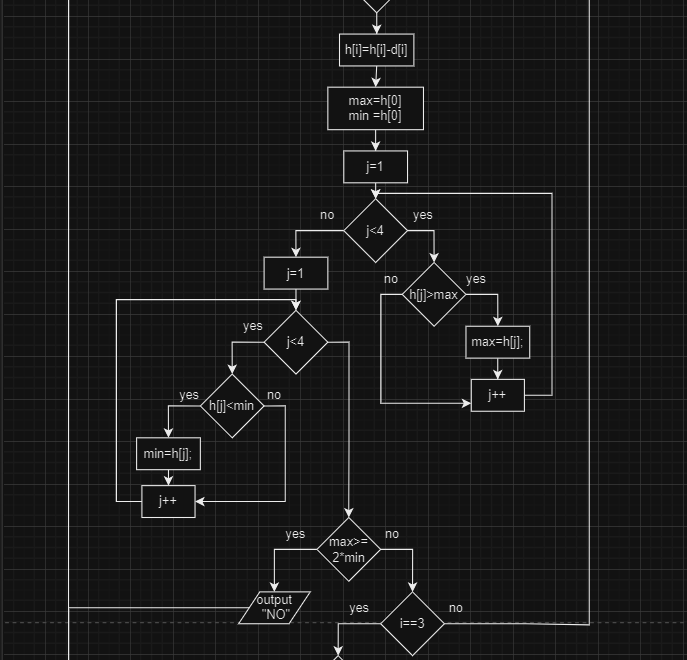
Програма №3 VNS Lab 2 Task 1

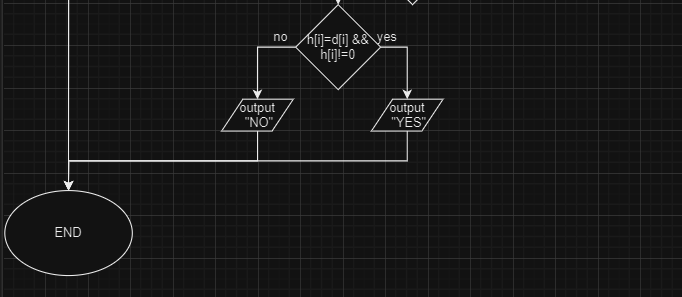
* Планований час на реалізацію : 50 хв



Програма №4 Algotester Lab 1 Task 2

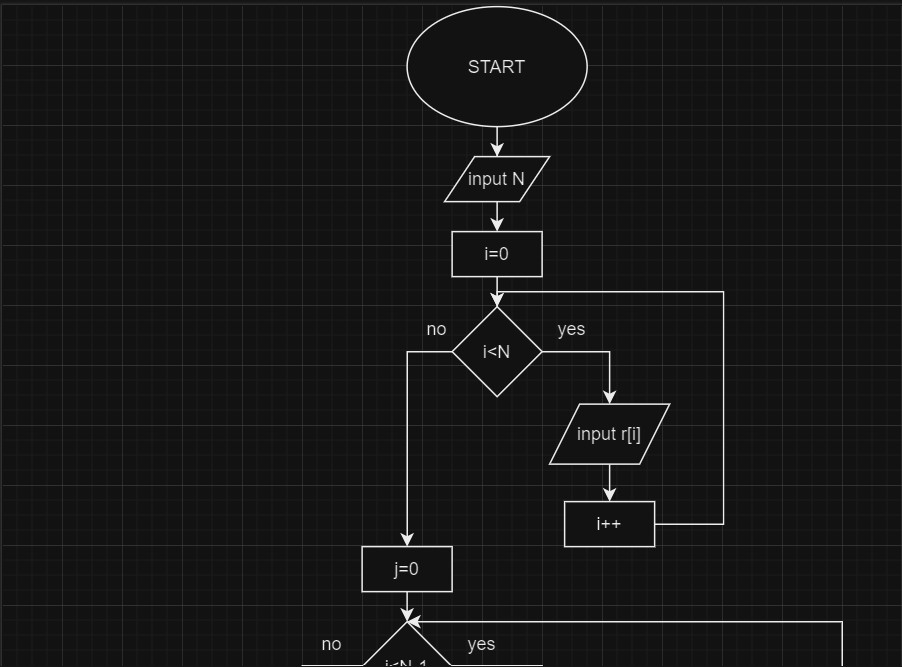


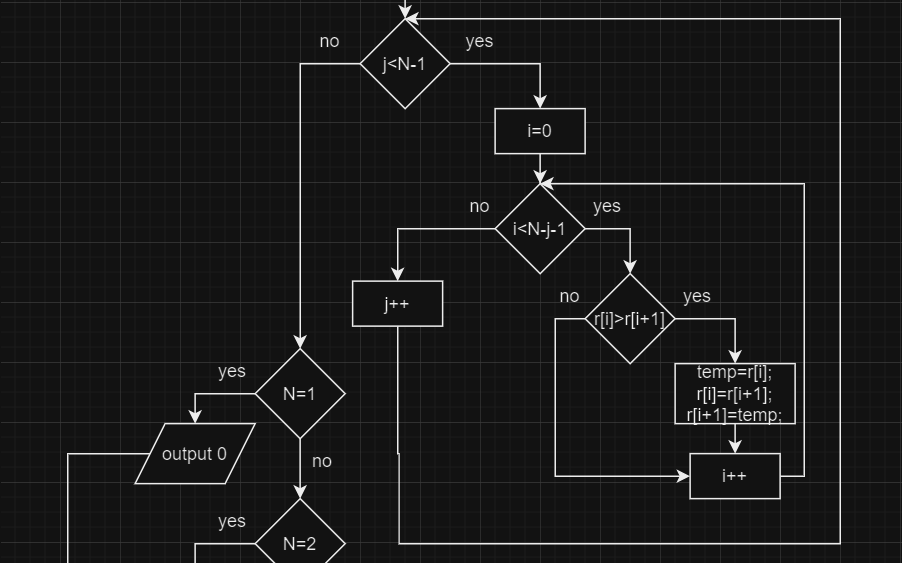


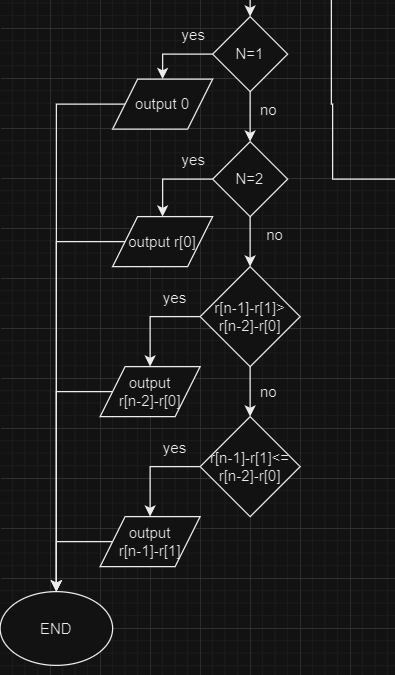


* Планований час на реалізацію : 1 год

Програма №5 Algotester Lab 2 Task 1

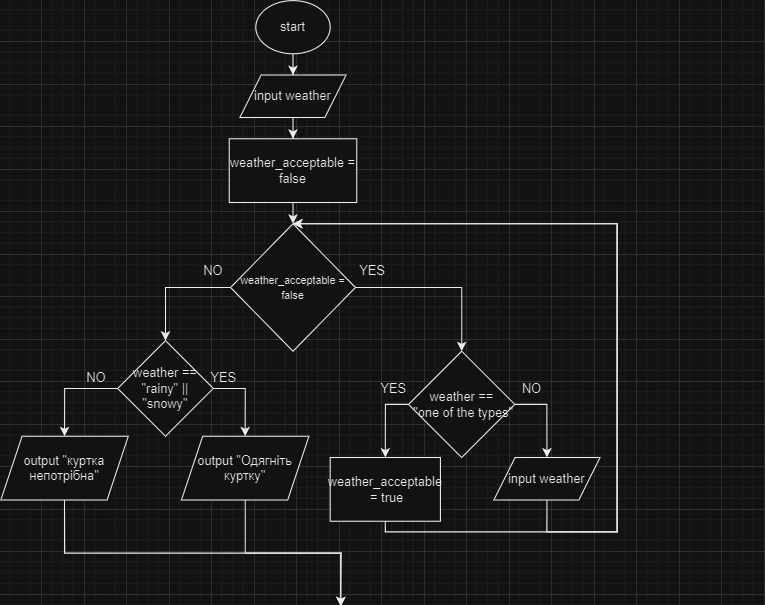


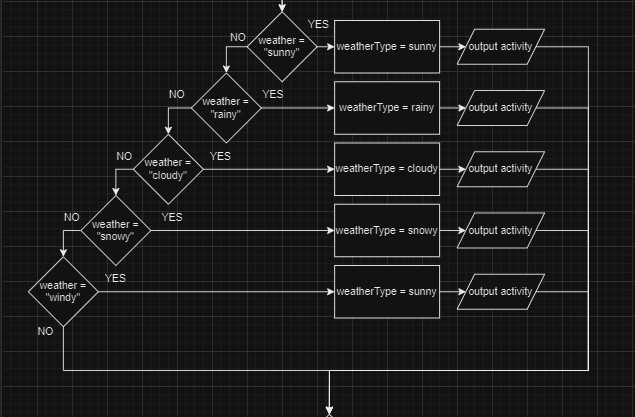


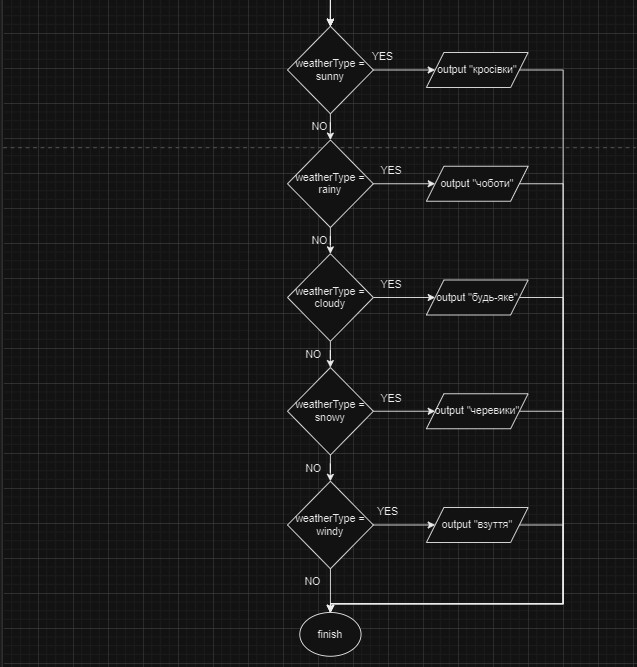


* Планований час на реалізацію : 1 год

Завдання №6 programming: Class Practice Task

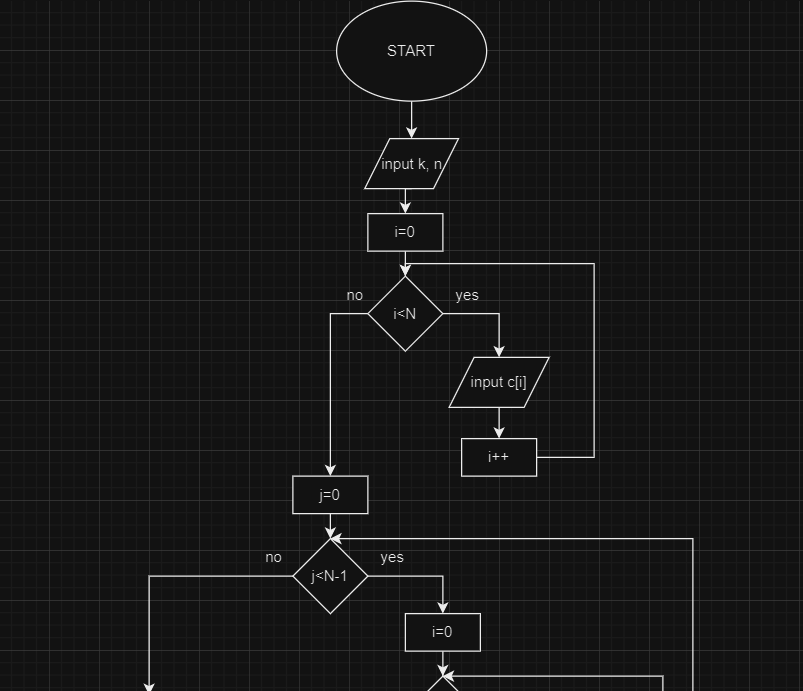


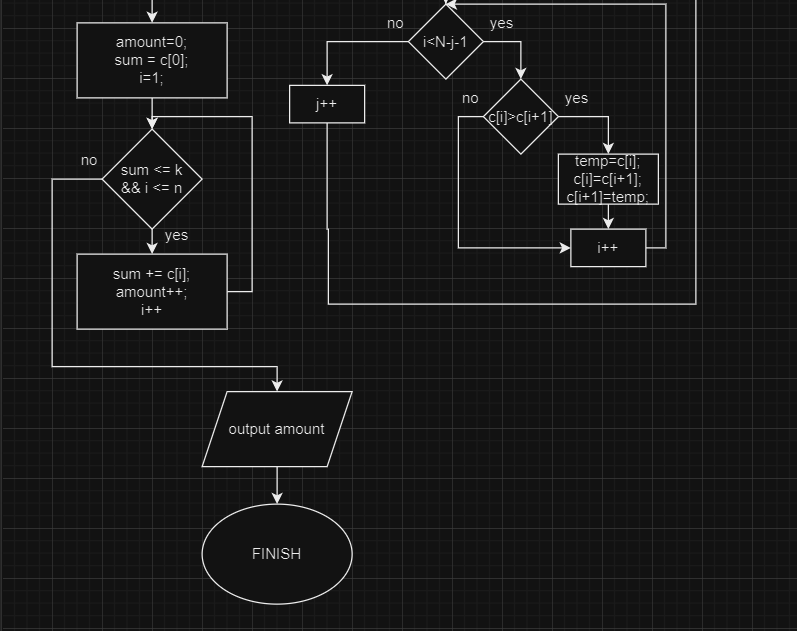




* Планований час на реалізацію : 1 пара

Завдання №7 programming: Self Practice Task





* Планований час на реалізацію : 30 хв

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

Використано налаштування з минулої лабораторної.

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

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 VNS Lab 2 Task 1

#include <iostream>

#include <math.h>

using *namespace* std;

// an=n!/3n^n

// an+1/an=(n/(n+1))^n

// an+1=an\*(n/(n+1))^n

*int* main(){

*int* n = 1;

*double* a, f=1, s=0, a\_next=1, e=0.001;

/\*

Щоб порахувати суму ряду з точністю, потрібно вивести наступний член цієї суми

тоді суму потрібно обраховувати, поки наступний член не стане менший за точність

\*/

while (a\_next >= e)

{

f = f \* n; //факторіал, починаючи з одиниці

a = f/(3\*pow(n,n));

s += a;

a\_next = a\*(pow(n,n)/pow(n+1,n));

n++;

cout << a << " ";

}

cout << "sum: " << s << endl;

return 0;

}

Завдання №4 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[3]<d[3] || 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 №5 Algotester Lab 2 Task 1

#include <iostream>

using *namespace* std;

*int* main(){

const *int* mx\_N = 100000;

*int* r[mx\_N];

*unsigned* *int* N;

//вводимо к-ть чисел у масиві

cin >> N;

//вводимо числа

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

cin >> r[i];

}

//відсортовуємо масив(в умові не сказано чи масив відсортований)

for (*int* j = 0; j < N; j++){

for (*int* i = 0; i < N - 1 - j; i++) {

if (r[i] > r[i+1]){

*int* temp = r[i];

r[i] = r[i+1];

r[i+1] = temp;

}

}

}

switch (N)

{

//якщо в масиві 1 елемент, то відкинувши його, ми отримаємо 0

case 1:

cout << 0;

break;

//якщо в масиві 2 елементи, то потрібно залишити мінімальний

case 2:

cout << r[0];

break;

//якщо в масиві більше елементів, то потрібно відкинути мінімальне або максимальне значення

default:

if (r[N-1] - r[1] > r[N-2] - r[0]){

cout << r[N-2] - r[0];

}

if (r[N-1] - r[1] < r[N-2] - r[0]){

cout << r[N-1] - r[1];

}

if (r[N-1] - r[1] == r[N-2] - r[0]){

cout << r[N-1] - r[1];

}

}

return 0;

}

Завдання №6 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 №7 programming: Self Practice Task

// algotester 0531

#include <iostream>

using *namespace* std;

*int* main(){

*int* c[100], n , k;

// вводимо к-ть грошей та атракціонів

cin >> k >> n;

// вводимо ціну на атракціони

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

cin >> c[i];

}

// відсортовуємо ціни по порядку зростання

for (*int* j = 0; j < n - 1; j++){

for (*int* i = 0; i < n - j - 1; i++) {

if (c[i] > c[i+1]){

*int* temp = c[i];

c[i] = c[i+1];

c[i+1] = temp;

}

}

}

// тоді перевіряємо для скількох атракціонів вистачить грошей

// сума починається з першого атракціону тоді перевіряємо чи вистачить грошей,

// якщо так, то зараховуємо атракціон і йдемо до наступного

*int* amount=0;

*int* sum = c[0];

*int* i = 1;

while(sum <= k && i <= n){

amount++;

sum += c[i];

i++;

}

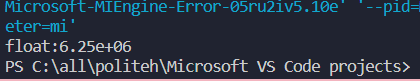
cout << amount << endl;

return 0;

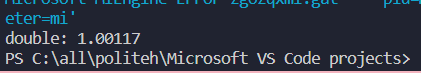
}

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

Завдання №1 Деталі по виконанню і тестуванню програми



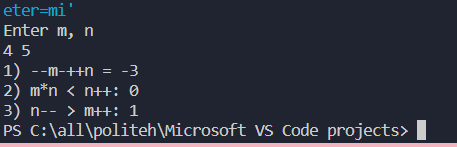
Виводить таке число через розмір розмір типу даних float та його точністю. Піднесення до квадрата скоріш за все призводить до втрати точності через обмежену кількість бітів у представленні чисел типу float



double має більші розмір і точність, тому виводить точніше значення

Час затрачений на виконання завдання: 20 хв

Завдання №2 Деталі по виконанню і тестуванню програми



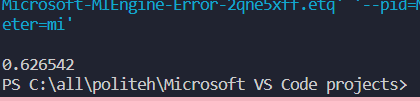
на першому етапі m=m-1=3, n=n+1=6 -> 3-6=-3

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

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

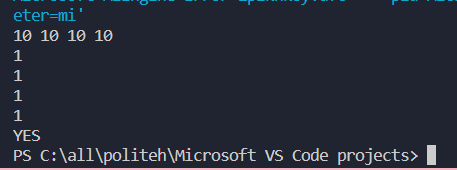
Час затрачений на виконання завдання: 20 хв

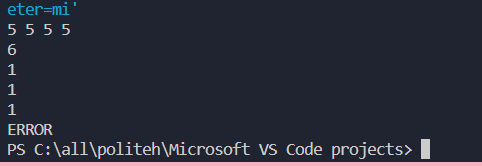
Завдання №3 Деталі по виконанню і тестуванню програми



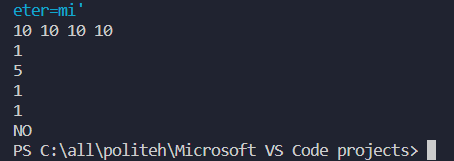
Час затрачений на виконання завдання: 45 хв

Завдання №4 Деталі по виконанню і тестуванню програми

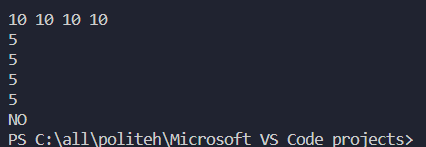




у цьому випадку довжина відпилювання більша за довжину



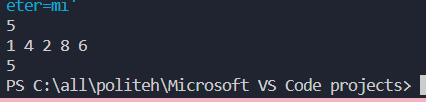
у цьому випадку після 2-го відпилювання друга ніжка буде вдвічі меншою за максимальну



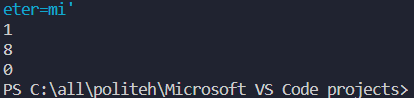
у цьому випадку після 1-го відпилювання друга ніжка буде вдвічі меншою за максимальну

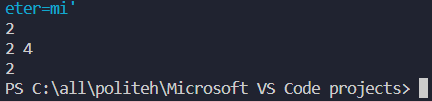
Час затрачений на виконання завдання: 1,5-2 год

Завдання №5 Деталі по виконанню і тестуванню програми



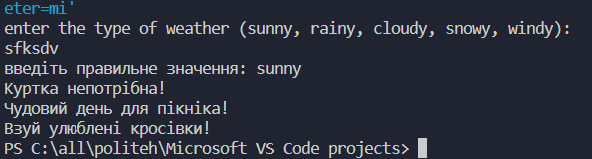
Відсортувавши масив маємо 1 2 4 6 8. Тоді, якщо відкинемо максимальний елемент різниця між новим максимальним і мінімальним буде 6-1=5, а якщо відкинути мінімальний, нова різниця буде 8-2=6. 5<6





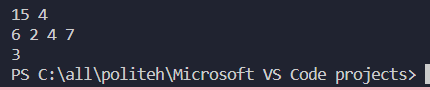
Час затрачений на виконання завдання 1-1,5 год

Завдання №6 Деталі по виконанню і тестуванню програми



Час затрачений на виконання завдання 1 пара

Завдання №7 Деталі по виконанню і тестуванню програми



Є 15 грн і 4 атракціони (6 грн, 2 грн, 4 грн, 7 грн). Спочатку Марічка піде на атракціон за 2 грн (сума = 2 грн < 15), тоді за 4 грн (сума = 6 грн < 15), тоді за 6 грн (сума = 12 < 15), а на 4 атракціон за 7 грн Марічці вже не вистачить грошей (сума = 17 > 15).

Час затрачений на виконання завдання 40 хв.

# **Висновки:**

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