Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

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

На тему: « Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Системи числення. Змінні. Константи. Типи даних. Розмір Типів Даних (Двійкова система). Ввід вивід. Базові операції та вбудовані функції. Коментарі.»

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

до:

ВНС Лабораторної Роботи № 1

Алготестер Лабораторної Роботи № 1

Практичних Робіт № 2

***Виконав:***

студент групи ШІ-11

Винницький Андрій Олегович

# **Тема роботи:**

# Аналіз та реалізація лінійних та розгалужених алгоритмів в контексті програмування: Умовні оператори, константи та змінні

# **Мета роботи:**

Розглянути основні аспекти лінійних та розгалужених алгоритмів у програмуванні, зосереджуючись на умовних операторах, константах і змінних. В рамках звіту вивчити та проаналізувати принципи побудови умов, використання змінних та констант для створення ефективних та логічних програм. Написати код для ілюстрації даних конструкцій в мові C++.

**Теоретичні відомості:**

1)  Теоретичні відомості з переліком важливих тем:

-  Тема №1: if, else if theory.

-  Тема №2: switch case theory.

2)  Індивідуальний план опрацювання теорії:

-  Тема №1: if, else if theory.

o  Джерела Інформації

▪️   <https://www.w3schools.com/cpp/cpp_conditions.asp>

o  Що опрацьовано:

▪️  Вивчено if/else if, написано коди з їх використанням.

o  Статус: Ознайомлений

o  Початок опрацювання теми: 25.10.2023

o  Завершення опрацювання теми: 7.11.2023

-  Тема №2: switch case theory.

o  Джерела Інформації:

<https://www.w3schools.com/cpp/cpp_switch.asp>

Що опрацьовано:

▪️  Вивчено switch case, написано коди з їх використанням.

o  Статус: Ознайомлений

Початок опрацювання теми: 25.10.2023

Завершення опрацювання теми: 5.11.2023

**Виконання роботи:**

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

#Task 2 - Requirements management and design activities

* Намалювати блок-схеми та оцінити завдання 3-7

#Task 3 - Lab# programming: VNS Lab 1 Task 1

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

#Task 4 - Lab# programming: VNS Lab 1 Task 2

* Обчислити значення виразів і пояснити їх результат

#Task 5 - Lab# programming: Algotester Lab 1 Task 1

* Написати код для задачі lab1v1

#Task 6 - Practice# programming: Class Practice Task

* Створити простий порадник щодо погоди

#Task 7  - Practice# programming:  Self Practice Task

* Реалізувати алгоритм Bucket sort

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

Task 2 - Requirements management and design activities

* Необхідно намалювати блок-схеми та оцінити завдання 3-7
* Запланований час 1 год.

Task 3 - Lab# programming: VNS Lab 1 Task 1

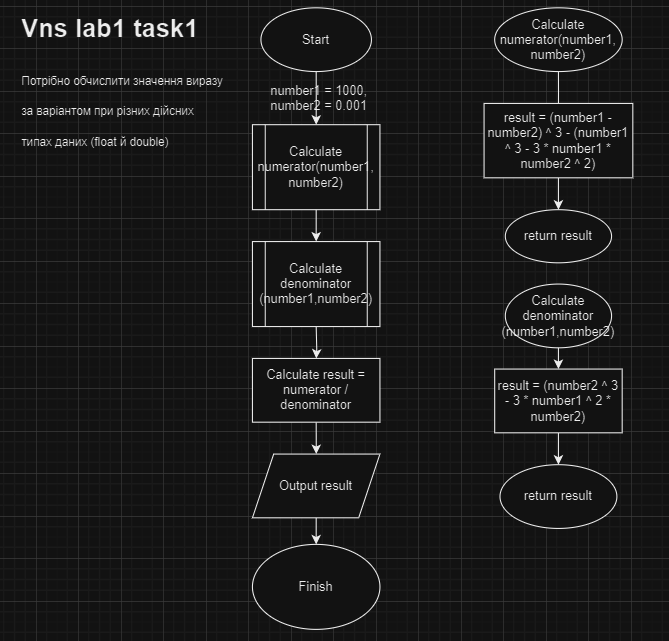


Рисунок :Лабораторна 1 завдання 1

* Запланований час 15 хв.
* Потрібно обчислити значення виразу за варіантом при різних дійсних типах даних (float й double)

Task 4 - Lab# programming: VNS Lab 1 Task 2

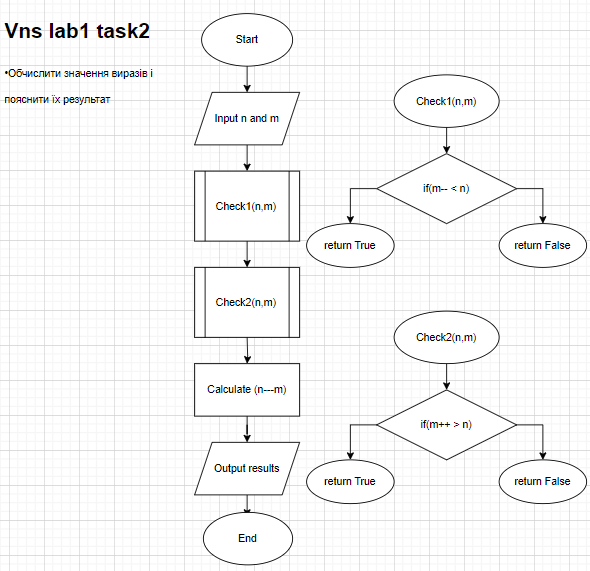


Рисунок :Лабораторна 1 завдання 2

* Запланований час 15 хв.
* Слід обчислити значення виразів і пояснити їх результат

Task 5 - Lab# programming: Algotester Lab 1 Task 1

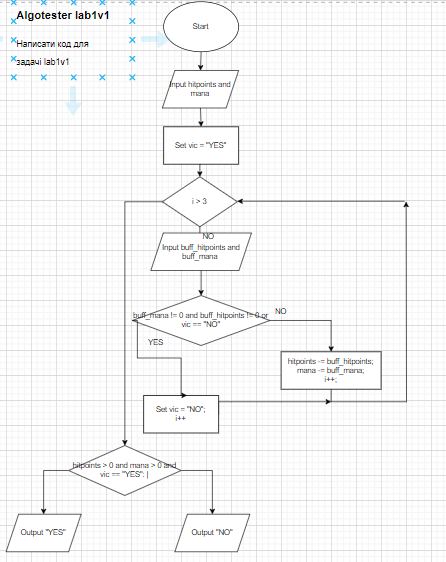


Рисунок : Лабораторна Алготестер

* Запланований час 20 хв.
* Потрібно написати код для задачі lab1v1

Task 6 - Practice# programming: Class Practice Task

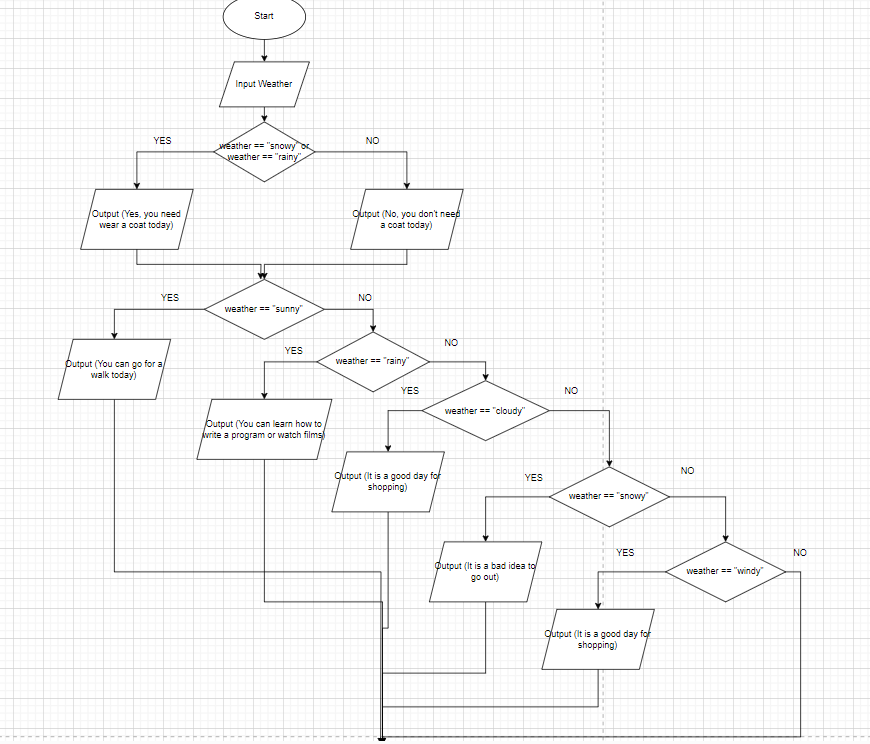
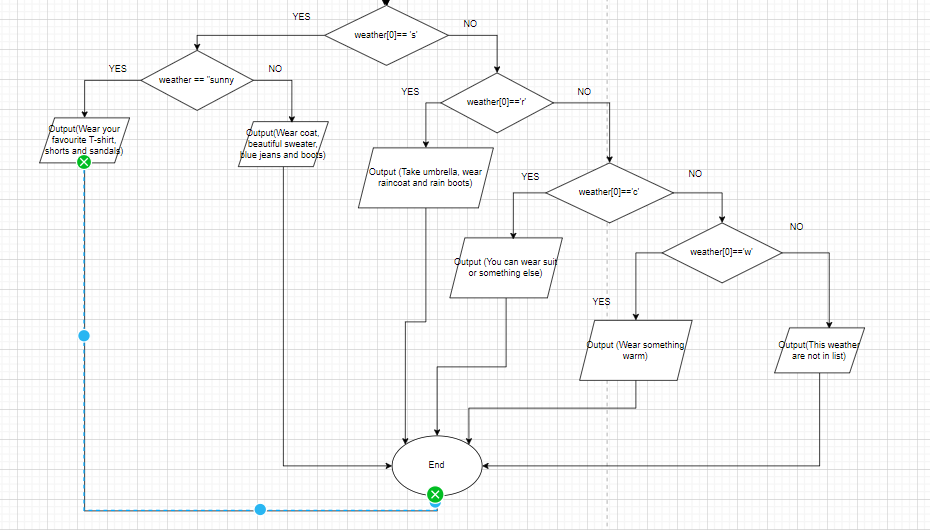
 

Рисунок : Завдання практичної

* Запланований час 20 хв.
* Необхідно створити простий порадник щодо погоди

Task 7  - Practice# programming:  Self Practice Task

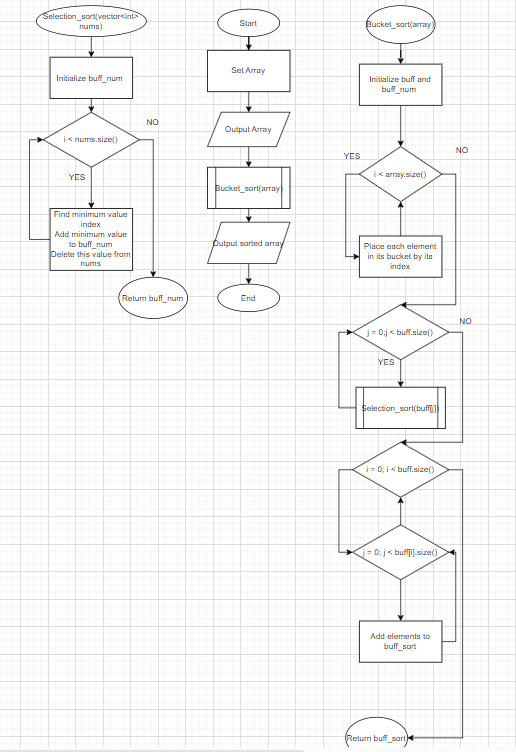


Рисунок : Selfpractice

* Запланований час 40 хв.
* Потрібно реалізувати алгоритм Bucket sort

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

Task 3 - Lab# programming: VNS Lab 1 Task 1 and Task 4 - Lab# programming: VNS Lab 1 Task 2

#include<iostream>  
using namespace std;  
  
  
double pow(double number, int count)  
{  
 if (count == 1)  
 {  
 return number;  
 }  
 return number \* pow(number,count - 1);  
}  
float pow\_f(float number, int count)  
{  
 if (count == 1)  
 {  
 return number;  
 }  
 return number \* pow(number,count - 1);  
}  
  
float numerator\_f(float num1 ,float num2)  
{  
 return (pow\_f(num1-num2, 3) - (pow\_f(num1, 3) - 3 \* num1 \* pow\_f(num2,2)));  
}  
float denominator\_f(float num1, float num2)  
{  
 return (pow\_f(num2, 3) -3 \* pow\_f(num1,2) \* num2);  
}  
double numerator(double num1 ,double num2)  
{  
 return (pow(num1-num2, 3) - (pow(num1, 3) - 3 \* num1 \* pow(num2,2)));  
}  
double denominator(double num1, double num2)  
{  
 return (pow(num2, 3) -3 \* pow(num1,2) \* num2);  
}  
string check1(double m ,double n)  
{  
 return (m-- < n) ? "True" : "False";  
}  
string check2(double m ,double n)  
{  
 return (m++ > n) ? "True" : "False";  
}  
double difference(double m, double n)  
{  
 return n---m;  
}  
  
int main()  
{  
 double number1 = 1000, number2 =0.001, m, n;  
 cout << "Enter m and n: ";  
 cin >> m;  
 cin >> n;  
 cout << "Result of first(float) task: "<< numerator\_f(number1, number2)/ denominator\_f(number1,number2)<<endl;  
 cout << "Result of first task: "<< numerator(number1, number2)/ denominator(number1,number2);  
 cout << "\nTask two:\n1)"<< difference(m , n);  
 cout << "\n2)"<< check1(m, n);  
 cout << "\n3)"<< check2(m,n);  
 return 0;  
}

Час витрачений : 15хв.

Task 5 - Lab# programming: Algotester Lab 1 Task 1

#include<iostream>  
using namespace std;  
  
  
int main()  
{  
 long long int hitpoints , mana, buff\_hitpoints, buff\_mana;  
 string vic = "YES";  
  
 cin>> hitpoints >> mana;  
 for (int i = 0; i < 3; i++){  
  
 cin>> buff\_hitpoints>> buff\_mana;  
 if (buff\_mana != 0 && buff\_hitpoints != 0 || vic == "NO"){  
 vic = "NO";  
 }  
  
 hitpoints -= buff\_hitpoints;  
 mana -= buff\_mana;  
  
 }  
 if (hitpoints > 0 && mana > 0 && vic == "YES")  
 {  
 cout<< vic;  
 }  
 else  
 {  
 cout<<"NO";  
 }

return 0;  
  
}

Час витрачений : 15хв.

Task 6 - Practice# programming: Class Practice Task

#include<iostream>  
using namespace std;  
  
  
int main() {  
 string weather;  
 cout << "Please enter the current weather\n- sunny\n- rainy\n- cloudy\n- snowy\n- windy\n: ";  
 cin >> weather;  
  
 if (weather == "snowy" || weather == "rainy") {  
 cout << "Yes, you need wear a coat today\n";  
 } else {  
 cout << "No, you don't need a coat today\n";  
 }  
  
  
 if (weather == "sunny")  
 {  
 cout << "You can go for a walk today\n";  
 }  
 else if (weather == "rainy")  
 {  
 cout << "You can learn how to write a program or watch films\n";  
 }  
 else if (weather == "cloudy")  
 {  
 cout << "It is a good day for shopping\n";  
 }  
 else if (weather == "snowy")  
 {  
 cout << "It is a bad idea to go out\n";  
 }  
 else if (weather == "windy") {  
 cout << "Try to stay on Earth, do not fly\n";  
 }  
  
  
  
 switch (weather[0]) {  
 case 's':  
 if(weather == "sunny")  
 {  
 cout << "Wear your favourite T-shirt, shorts and sandals\n";  
 }  
 else  
 {  
 cout << "Wear coat, beautiful sweater, blue jeans and boots\n";  
 }  
 break;  
 case 'r':  
 cout << "Take umbrella, wear raincoat and rain boots\n";  
 break;  
 case 'c':  
 cout << "You can wear suit or something else\n";  
 break;  
 case 'w':  
 cout << "Wear something warm\n";  
 break;  
 default:  
 cout << "This weather are not in list\n";  
 break;  
 }  
  
  
 return 0;  
}

Час витрачений : 20 хв.

Task 7  - Practice# programming:  Self Practice Task

#include <iostream>  
#include <vector>  
#include <algorithm>  
#include <ctime>  
using namespace std;  
  
int min\_vec\_idx(vector<int> nums)  
{  
 int min = nums[0];  
 int min\_idx = 0;  
 for(int i = 0; i < nums.size(); i++)  
 {  
 if(min > nums[i])   
 {  
 min = nums[i];  
 min\_idx = i;  
 }  
 }  
 return min\_idx;  
}  
  
  
vector<int> selection\_sort(vector<int> nums)  
{  
 vector<int> buff\_num;  
 for(int i = nums.size();i >= 0 && nums.size() != 0; i--)  
 {  
 int min = min\_vec\_idx(nums);  
 buff\_num.push\_back(nums[min]);  
 nums.erase(nums.begin() + min );  
 }  
 return buff\_num;  
}  
  
  
  
vector<int> bucket\_sort(vector<int> numbers)  
{  
 int max = \*max\_element(numbers.begin(),numbers.end());  
 vector<vector<int>> buff(max/10 + 1);  
 vector<int> buff\_sort;  
 for (int i = 0;i < numbers.size(); i++)  
 {  
 buff[numbers[i]/10].push\_back(numbers[i]);  
 }  
 for (int j = 0;j < buff.size();j++)  
 {  
 buff[j]= selection\_sort(buff[j]);  
 }  
 for(int i = 0; i < buff.size(); i++)  
 {  
 for(int j = 0; j < buff[i].size(); j++)  
 {  
 buff\_sort.push\_back(buff[i][j]) ;  
 }  
 }  
 return buff\_sort;  
  
  
  
}  
void print\_array(vector<int> numbers)  
{  
 for(int i = 0; i < numbers.size();i++)  
 {  
 cout<< numbers[i]<<" ";  
 }  
}  
  
int main()  
{  
 srand(time(0));  
 vector<int> numbers = {};  
 for(int i = 0; i < 10;i++)  
 {  
 numbers.push\_back(rand() % 100);  
 }  
 cout<<"Array: ";  
 print\_array(numbers);  
 numbers = bucket\_sort(numbers);  
 cout<<"\nSorted array: ";  
 print\_array(numbers);  
  
 return 0;  
}

Час витрачений : 45хв.

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

Task 3 - Lab# programming: VNS Lab 1 Task 1 and Task 4 - Lab# programming: VNS Lab 1 Task 2

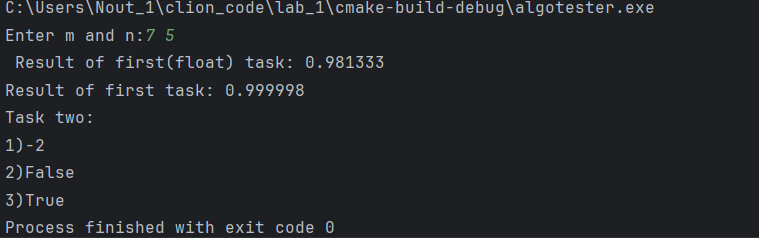
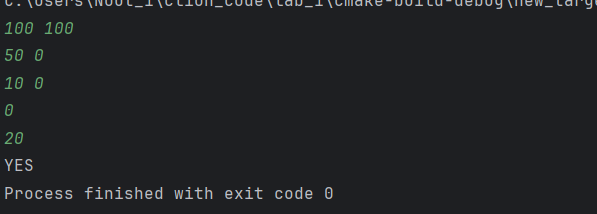


Рисунок :Код обчислює значення виразу за варіантом при різних дійсних типах даних (float й double) і обчислює значення виразів

Task 5 - Lab# programming: Algotester Lab 1 Task 1



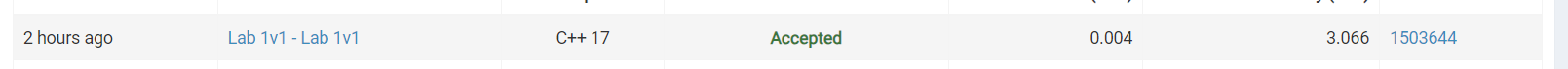
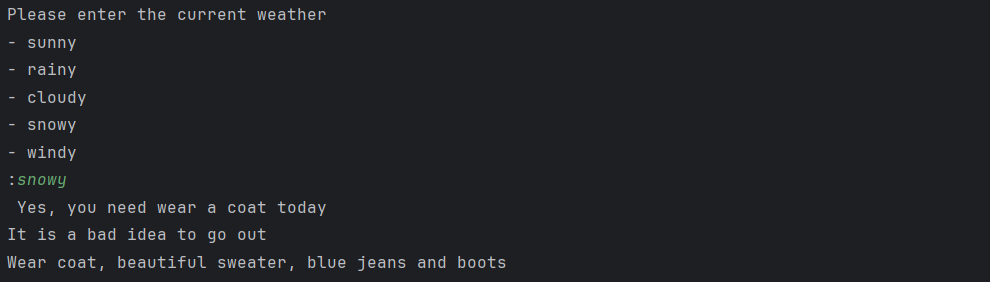


Рисунок : Результати лабораторної з алготестеру

Код виконує задачу lab1v1

Task 6 - Practice# programming: Class Practice Task



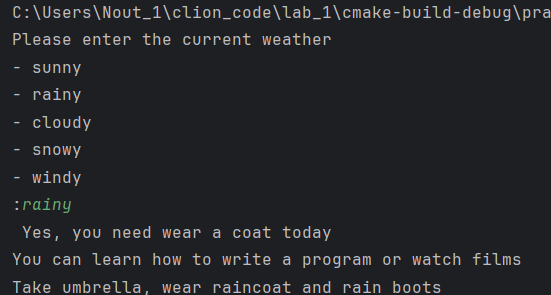


Рисунок : Поради щодо погоди

Task 7  - Practice# programming:  Self Practice Task



Рисунок :Результат Selfpractice

Відсортовує масив алгоритмом bucket

Час затрачений на виконання завдань : 2.5 год

Час витрачений на звіт 3 год.

# **Висновки:**

За цей епік я навчився вивчив лінійні та розгалужені алгоритми, умовні оператори(if, else, else if, switch case), константи, змінні і навчився їх застосовувати, пишучи код на мові програмування C++