Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №4**

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

***з розділу***: «Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки.»

***Виконала:***

студентка групи ШІ-14

Павлюх Вікторія Ігорівна

# **Тема роботи:**

Розглянути прості структури даних.Одновимірні масиви , двовимірні масиви.Алгоритми обробки.

# **Мета роботи:**

Ознайомитися з одновимірними та двовимірними у С++. Опрацювати та застосувати алгоритми обробки для масивів.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Структури даних.
* Тема №2: Статичний масив.
* Тема №3: Динамічний масив.
* Тема №4: Одновимірний масив.
* Тема №5: Двовимірний масив.
* Тема №6: Алгоритми обробки.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Структури даних.
  + Джерела Інформації
    - Статті:<https://acode.com.ua/urok-64-struktury/>
  + Що опрацьовано:
    - Структури даних їхні типи та їхні відмінності.
  + Статус: Ознайомлена
  + Витрачений час: тиждень
* Тема №2: Статичний масив.
  + Джерела Інформації:
    - Статті: <https://purecodecpp.com/uk/archives/3205>
  + Що опрацьовано:
    - Реалізація статичного масиву та дії над ним.
  + Статус: Ознайомлена
  + Витрачений час: два тижні
* Тема №3: Динамічний масив.
  + Джерела Інформації:
    - Статті: 1. <https://acode.com.ua/urok-90-dynamichni-masyvy/>

2. <https://acode.com.ua/urok-99-vvedennya-v-std-vector/>

* + Що опрацьовано:
    - Реалізація динамічного масиву та дії над ним.
  + Статус: Ознайомлена
  + Витрачений час: два тижні
* Тема №4 Одновимірний масив.
  + Джерела Інформації:
    - Стаття: [1.https://www.bestprog.net/uk/2017/03/01/%D0%BC%D0%B0%D1%81%D0%B8%D0%B2%D0%B8-%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B0-1-%D0%B2%D0%B8%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F-%D0%BC%D0%B0%D1%81%D0%B8%D0%B2%D1%83-%D0%BE%D0%B4/](https://www.bestprog.net/uk/2017/03/01/%D0%BC%D0%B0%D1%81%D0%B8%D0%B2%D0%B8-%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B0-1-%D0%B2%D0%B8%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F-%D0%BC%D0%B0%D1%81%D0%B8%D0%B2%D1%83-%D0%BE%D0%B4/)
  + Що опрацьовано:
    - Реалізацію одновимірного масиву.
  + Статус: Ознайомлена
  + Витрачений час: тиждень
* Тема №5 Двовимірний масив.
  + Джерела Інформації:
    - Cтатті: <https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
  + Що опрацьовано:
    - Реалізацію двовимірного масиву, та його відмінність від одновимірного масиву.
  + Статус: Ознайомлена
  + Витрачений час: тиждень
* Тема №6 Алгоритми обробки.
  + Джерела Інформації:
    - Статті:1.<https://cherto4ka.xyz/2020/01/21/%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D1%96-%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%97-%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B8-%D0%BE%D0%B4%D0%BD%D0%BE%D0%B2%D0%B8%D0%BC%D1%96%D1%80%D0%BD%D0%B8/>

2.<https://acode.com.ua/urok-80-sortuvannya-masyviv-metodom-vyboru/>

3.<https://acode.com.ua/urok-207-algorytmy-stl/>

* + Що опрацьовано:
    - Бібліотеку STL та її алгоритми.
  + Статус: Ознайомлена
  + Витрачений час: два тижні.

# **Виконання роботи:**

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

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

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

## **4. Результати виконання завдань, тестування**

**Завдання №2 VNS Lab 4 - Task 1**

- Варіант: 7

- Деталі завдання:

1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.

2) Роздрукувати отриманий масив.

3) Знищити останній елемент. який дорівнює 0.

4) Додати після елемента масиву із заданим індексом елемент зі значенням 100.

5) Роздрукувати отриманий масив.

- Блок-схема: не робила.

- Код програми:

*#include* <vector>

*#include* <iostream>

*#include* <cstdlib>

using namespace std;

int main()

{

int randomDigits\_size;

cout << "Введіть розмір масиву: ";

cin >> randomDigits\_size;

vector<int> randomDigits;

for (int i = 0; i < randomDigits\_size; i++)

{

randomDigits.push\_back(rand());

cout << randomDigits[i]<< " " ;

}

cout <<endl;

if(randomDigits[randomDigits\_size-1]==0){

randomDigits.pop\_back();

cout << "Масив, після видалення останнього елемента: ";

for(int i = 0; i < randomDigits\_size; ++i)

cout << randomDigits[i] << " ";

}

int index;

cout << "Введіть індекс, після якого потрібно вставити новий елемент: ";

cin >> index;

if (index >= 0 && index < randomDigits\_size)

{

randomDigits.insert(randomDigits.begin() + index + 1, 100);

cout << "Масив після вставки нового елемента зі значенням 100: ";

for (int i = 0; i < randomDigits.size(); ++i)

cout << randomDigits[i] << " ";

cout << endl;

}

else

{

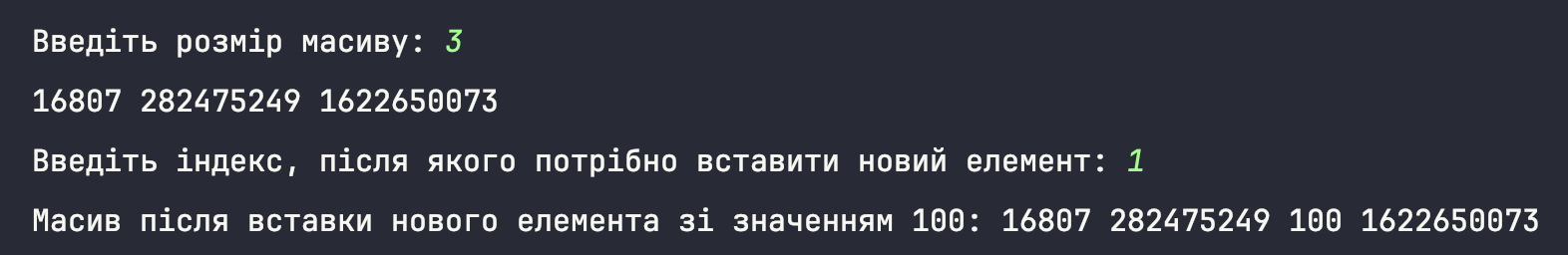
cout << "Некоректний індекс для вставки." << endl;

}

return 0;

}

- Результати:

**

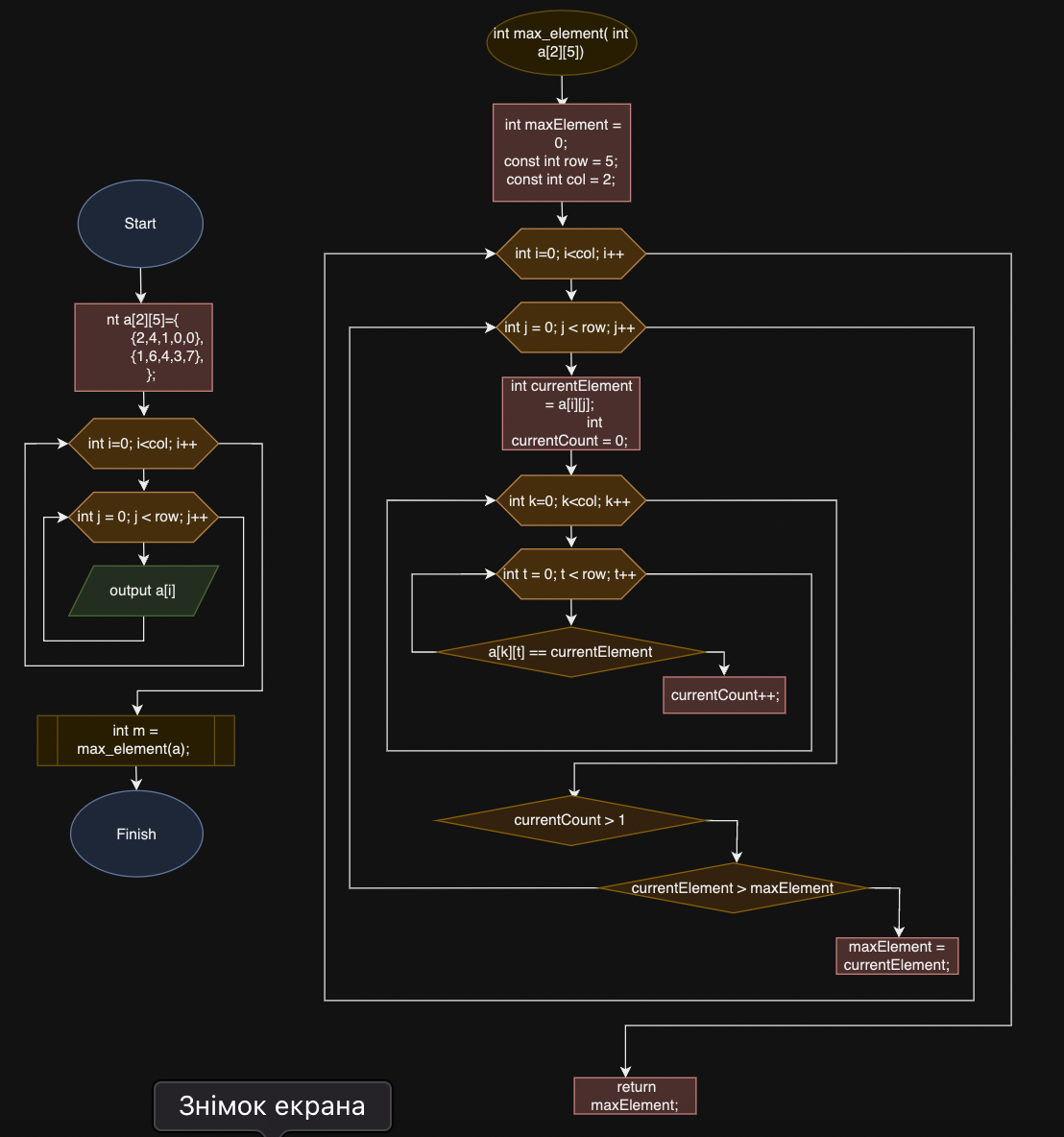
*Рис.1.Результати тестування.*

**Завдання №2 VNS Lab 5 - Task 1**

- Варіант 21

- Деталі завдання: Використовуючи функції, розв’язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр. Знайти максимальний з повторюваних елементів двовимірного масиву.

- Блок-схема:



*Рис.2.Блок-схема до 5 лабораторної.*

- Код програми:

*#include* <iostream>

*#include* <string>

using namespace std;

const int row = 5;

const int col = 2;

int max\_element( int *a*[2][5]){

int maxElement = 0;

for(int i=0; i<col; i++) {

for (int j = 0; j < row; j++) {

int currentElement = *a*[i][j];

int currentCount = 0;

for(int k=0; k<col; k++) {

for (int t = 0; t < row; t++) {

if (*a*[k][t] == currentElement) {

currentCount++;

}

}

}

if (currentCount > 1) {

if (currentElement > maxElement) {

maxElement = currentElement;

}

}

}

}

return maxElement;

}

int main() {

int a[2][5]={

{2,4,1,0,0},

{1,6,4,3,7},

};

for(int i=0; i<col; i++) {

for (int j = 0; j < row; j++) {

cout << a[i][j] << " ";

}

cout <<endl;

}

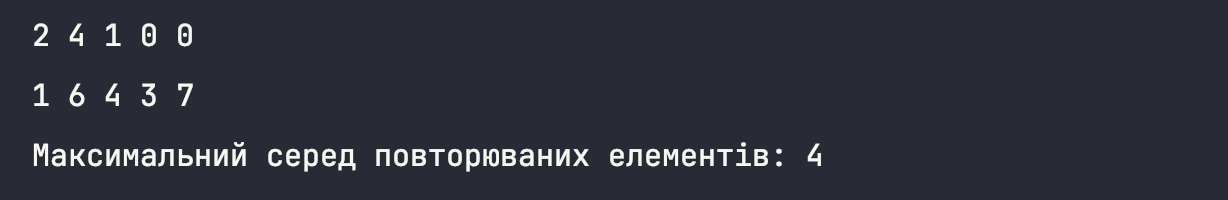
int m = max\_element(a);

cout << "Максимальний серед повторюваних елементів: " <<m<<endl;

return 0;

}

- Результати:

**

*Рис.3.Результат тестування.*

Програма виводить найбільше серед повторюваних елементів двовимірного масиву.

**Завдання №3 Algotester Lab 2**

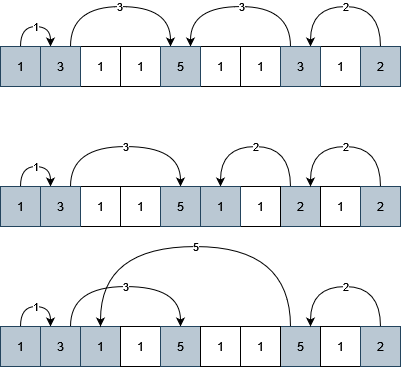
- Варіант: 3

- Деталі завдання: Вам дано масив цілих чисел розміром N, на першій та останній клітинці розміщено по дрону. Вони одночасно взлітають. На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться. Тобто лівий дрон у першу секунду з клітинки з індексом 1 перелетить у клітинку з індексом a[1], тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону. Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якщо 2 дрони опиняються в одній клітинці - ви виводите **Collision**.

Якщо лівий дрон опиниться справа від правого - це **Miss**

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках a[i] та a[i+1] - виведіть **Stopped.** Врахуйте, що перевіряти треба також до взльоту.



*Рис.4.Візуалізація завдання.*

- Блок-схема: не робила.

- Код програми:

*#include* <iostream>

*#include* <vector>

using namespace std;

int main(){

int N, pos1, pos2, i1, i2;

*string* result="-";

cin >> N;

int \*arr = new int[N];

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

cin >> arr[i];

}

if(N==2){

result="Stopped";

} else if (N==1){

result="Collision";

}

i1 = 0;

i2 = N-1;

for(; i1<N && i2>i1-1 && i2>i1+1; ){

i1+=arr[i1];

i2-=arr[i2];

}

if(i1>i2){

result="Miss";

}else if(i1==i2){

result="Collision";

}else if(i1==i2-1 || i1==i2+1){

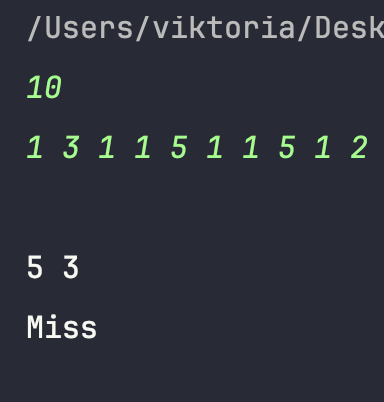
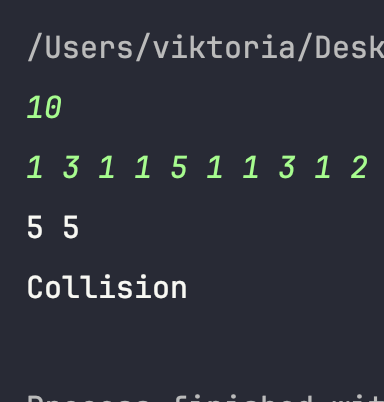
result="Stopped";

}

cout << i1 + 1 << ' '<< i2 + 1 << endl<< result << endl;

}

- Результати:



*Рис.5-7.Результати тестування.*

**Завдання №4 Algotester Lab 3**

- Варіант: 2

- Деталі завдання: Вам дано 2 масиви розміром N та M. Значення у цих масивах унікальні. Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

- Блок-схема: не робила.

- Код програми:

*#include* <iostream>

using namespace std;

int main() {

int lengthN, lengthM, count=0;

cin>>lengthN;

int \*N=new int[lengthN];

for(int i=0; i<lengthN; i++){

cin>>N[i];

}

cin>>lengthM;

int \*M=new int[lengthM];

for(int i=0; i<lengthM; i++){

cin>>M[i];

}

for(int i=0; i<lengthM; i++){

for(int j=0; j<lengthN; j++){

if(M[i]==N[j]){

count++;

}

}

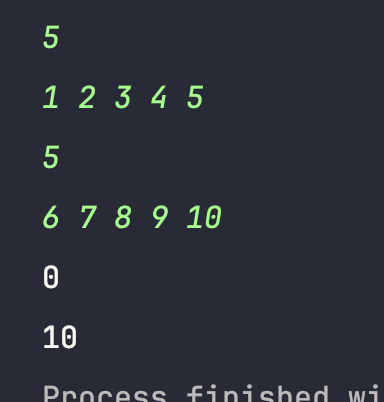
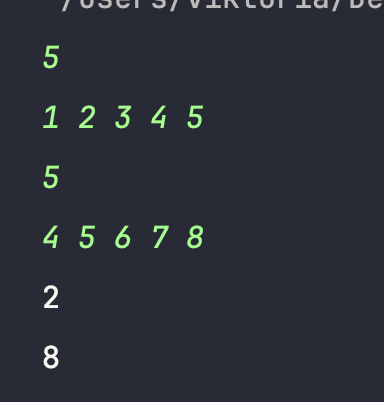
}

cout<<count<<endl;

cout<<(lengthN+lengthM-count);

}

- Результати:



*Рис.8-9. Результати тестування.*

**Завдання №5 Class Practice Work**

- Варіант: Без варіанту.

- Деталі завдання: Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

### Вимоги:

1. Визначення функції:
   1. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
   1. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
   1. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
   2. *bool isPalindrome*(ціле число);
4. Рекурсія:
   1. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

- Блок-схема: не робила

- Код програми:

*#include* <iostream>

*#include* <string>

using namespace std;

bool isPalindrome(const *string*& *str*, int *start*, int *end*){

if(*start*>=*end*){

return true;

}

if (*str*[*start*] != *str*[*end*-1]) {

return false;

}

return isPalindrome(*str*, *start* + 1, *end* - 1);

}

bool isPalindrome(int *length*, int *number*[]) {

for (int i = 0; i < *length* / 2; i++) {

if (*number*[i] != *number*[*length* - i - 1]) {

return false;

}

}

return true;

}

int main() {

*string* word;

cout << "Введіть рядок: ";

cin >> word;

if (isPalindrome(word, 0, word.length())== true) {

cout << "Це паліндром." << endl;

} else {

cout << "Це не паліндром." << endl;

}

int length;

cout << "Введіть кількість чисел: ";

cin>>length;

cout << "Введіть число: ";

int \*number = new int[length];

for(int i=0; i<length; i++){

cin>>number[i];

}

if (isPalindrome(length, number) == true) {

cout << "Це паліндром." << endl;

} else {

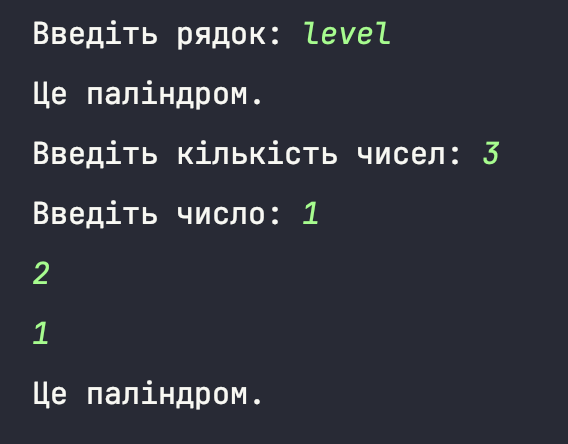
cout << "Це не паліндром." << endl;

}

return 0;

}

- Результати:



*Рис.10.Результат тестування.*

**Завдання №6 Self Practice Work**

- Задача з алготестеру 0021 - Депутатські гроші.

- Деталі завдання: Тож уявіть собі депутата, у якого є необмежена кількість купюр усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Він хоче придбати подарунок для своєї тещі, що коштує n гривень. Незважаючи на те, що наш герой-олігарх є нескінченно багатим, він також є нескінченно скупим та педантичним. Саме тому він хоче оплатити покупку готівкою без решти. Яка мінімальна кількість купюр йому для цього знадобиться?

- Блок-схема: не робила.

- Код програми:

*#include* <iostream>

using namespace std;

int main() {

long n;

cin>>n;

int m=0;

while(n>0){

while ((n - 500) >= 0) {

n-=500;

m++;

}

while((n-200)>=0){

n-=200;

m++;

}

while((n-100)>=0){

n-=100;

m++;

}

while((n-50)>=0){

n-=50;

m++;

}

while((n-20)>=0){

n-=20;

m++;

}

while((n-10)>=0){

n-=10;

m++;

}

while((n-5)>=0){

n-=5;

m++;

}

while((n-2)>=0){

n-=2;

m++;

}

while((n-1)>=0){

n-=1;

m++;

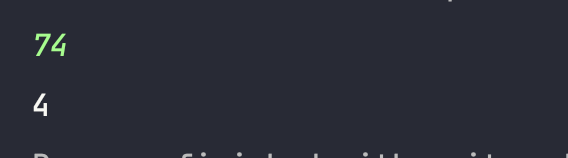
}

cout<<m;

}

}

- Результати:



*Рис.11. Результат тестування.*

**Висновки:**

Ознайомилася з різними масивами, структурами даних, з алгоритмами їхньої обробки. Під час виконання лабораторних задач застосувала їх.