

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра систем штучного інтелекту



**Звіт**  
**про виконання лабораторних та практичних робіт блоку № 4**  
**з дисципліни: «Основи програмування»**

На тему: «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання.  
Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та  
робота з масивами та структурами.»

**з дисципліни: «Основи програмування»**

до:

ВНС Лабораторної Роботи №4  
ВНС Лабораторної Роботи №5  
Алготестер Лабораторної Роботи №2  
Алготестер Лабораторної Роботи №3  
Практичних Робіт до блоку №4

**Виконав:**

Студент групи ІІІ-11

Мартин Максим Ігорович

**Тема роботи:** Ознайомлення з одновимірними та двовимірними масивами, їх оголошенням, ініціалізацією та використанням у програмах C++. Вивчення взаємодії між масивами та вказівниками, особливостей їх використання в контексті функцій, а також принципів роботи з посиланнями. Розгляд різниці між статичними та динамічними масивами, а також алгоритмів роботи з динамічною пам'яттю, використання операторів new та delete. Вивчення структур даних, включаючи вкладені структури та об'єднання (union), їх оголошення та практичне застосування. Аналіз алгоритмів обробки масивів і структур, їх інтеграції в більш складні алгоритми та застосування у вирішенні різних завдань.

**Мета роботи:** Опанувати методи створення та ініціалізації одновимірних і двовимірних масивів, виконання основних операцій, таких як індексація, обхід і маніпуляції з даними за допомогою циклів та функцій. Навчитися використовувати вказівники для доступу до елементів масивів, розуміти їхню арифметику та зв'язок з посиланнями. Дослідити різницю між вказівниками та посиланнями, особливо в контексті передачі параметрів у функції. Засвоїти динамічне виділення пам'яті та управління нею за допомогою операторів new і delete, створювати та використовувати динамічні масиви. Зрозуміти основи роботи зі структурами даних, навчитися оголошувати та використовувати вкладені структури та об'єднання, моделювати складні об'єкти. Ознайомитися з практичними прикладами алгоритмів пошуку та сортування в масивах, навчитися їх застосовувати для обробки даних у різних структурах. Розвинути вміння створювати, аналізувати та оптимізувати алгоритми для роботи з масивами та структурами даних у контексті вирішення практичних задач.

### **Теоретичні відомості:**

Теоретичні відомості з переліком важливих тем:

- Тема №1: Класи пам'яті в C++.
- Тема №2: Вступ до Масивів і Вказівників.
- Тема №3: Вказівники та Посилання.
- Тема №4: Динамічні Масиви.
- Тема №5: Структури Даних.
- Тема №6: Вкладені Структури.
- Тема №7: Алгоритми обробки та робота з Масивами та структурами.

Використовув :

[Джерала 1](#) , [Джерала 2](#) , [Джерала 3](#) , [Джерала 4](#) , [Джерала 5](#) , [Джерала 6](#)

## Виконання роботи:

### 1. VNS - Лабораторна робота № 4 , 11 варіант

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити 5 останніх елементів масиву.
- 4) Додати в початок масиву 3 елементи зі значенням  $M[I+1]+2$ .
- 5) Роздрукувати отриманий масив.

```
1 // 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
2 // 2) Знищити 5 перших елементів масиву.
3 // 3) Додати в кінець масиву 3 нових елементи.
4 #include <iostream>
5 #include <cstdlib>
6 #include <ctime>
7
8 using namespace std;
9
10 int main() {
11     const int initialSize = 10;
12     int arr[initialSize];
13     int currentSize = initialSize;
14
15     srand(time(0));
16
17     for (int i = 0; i < currentSize; i++) {
18         arr[i] = rand() % 100;
19     }
20
21     cout << "Initial array: ";
22     for (int i = 0; i < currentSize; i++) {
23         cout << arr[i] << " ";
24     }
25     cout << endl;
26
27     if (currentSize >= 5) {
28         currentSize -= 5;
29     }
30
31     const int additionalSize = 3;
32     int newArr[initialSize + additionalSize];
33
34     for (int i = 0; i < additionalSize; i++) {
35         newArr[i] = (i + 1 < currentSize ? arr[i + 1] : 0) + 2;
36     }
37
38     for (int i = 0; i < currentSize; i++) {
39         newArr[i + additionalSize] = arr[i];
40     }
41
```

```
38     for (int i = 0; i < currentSize; i++) {
39         newArr[i + additionalSize] = arr[i];
40     }
41
42     currentSize += additionalSize;
43
44     cout << "Array after changes: ";
45     for (int i = 0; i < currentSize; i++) {
46         cout << newArr[i] << " ";
47     }
48     cout << endl;
49
50     return 0;
51 }
```

Initial array: 13 87 29 59 44 74 9 82 91 1

Array after changes: 89 31 61 13 87 29 59 44

PS C:\Users\Maks\Documents\ai\_programming\_playground\_2024\ai\_11\maksym\_martyn\epic\_4> []

## 2. VNS - Лабораторна робота № 5 , 11 варіант

```
1  #include <iostream>
2  using namespace std;
3
4  bool isAscending(int row[], int column){
5      for(int i = 0; i < column-1;i++){
6          if(row[i] > row[i+1]){
7              return false;
8          }
9      }
10     return true;
11 }
12
13 bool isDescending(int row[], int column){
14     for(int i = 0; i < column-1;i++){
15         if(row[i] < row[i+1]){
16             return false;
17         }
18     }
19     return true;
20 }
21
22 void sortAscending (int row[], int column){
23     int temp;
24     for (int i = 0; i < column-1; i++){
25         for(int j = 0;j < column - i - 1; j++){
26             if(row[j] > row[j+1]){
27                 temp = row[j];
28                 row[j] = row[j+1];
29                 row[j+1] = temp;
30             }
31         }
32     }
33 }
34
```

```
35 int main (){
36     const int rows = 5;
37     const int columns = 5;
38     int arr[rows][columns] = {
39         {5, 3, 1, 2, 4},
40         {10, 9, 8, 7, 6},
41         {5, 4, 3, 2, 1},
42         {1, 2, 3, 4, 5},
43         {44, 41, 54, 14, 4}
44     };
45
46     cout << "Initial array: " << '\n';
47     for(int i = 0; i < rows; i++){
48         for(int j = 0;j < columns; j++){
49             cout << arr[i][j] << ' ';
50         }
51         cout << '\n';
52     }
53
54     for(int i = 0; i < rows ;i++){
55         if(!isAscending(arr[i],columns) && !isDescending(arr[i],columns)){
56             sortAscending(arr[i],columns);
57         }
58     }
59     cout << "Array after changing " << '\n';
60     for(int i = 0; i < rows; i++){
61         for(int j = 0; j < columns; j++){
62             cout << arr[i][j] << ' ';
63         }
64         cout << '\n';
65     }
66
67     return 0;
68 }
```

```
Initial array:
5 3 1 2 4
10 9 8 7 6
5 4 3 2 1
1 2 3 4 5
44 41 54 14 4
Array after changing
1 2 3 4 5
10 9 8 7 6
5 4 3 2 1
1 2 3 4 5
4 14 41 44 54
```

### 3.Practice work

Перевірка чи слово або число є паліндромом

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

#### Мета Задачі:

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  bool isPalindrome (const string &str){
6      int start = 0;
7      int end = str.length()-1;
8      while (start<end)
9      {
10         if (str[start]!=str[end])
11         {
12             return false;
13         }
14         start++;
15         end--;
16     }
17     return true;
18 }
19
20 bool isPalindrome (int num){
21     string forNum = to_string(num);
22     return isPalindrome(forNum);
23 }
24
25 void Print (bool &result){
26     if(result){
27         cout << "Is Palindrome";
28     } else {
29         cout << "Is not Palindrome";
30     }
31 }
32
```

```

33 int main(){
34     int option;
35     string str;
36     bool result;
37     int num;
38
39     cout << "Enter option: 1 - for string , 2 - for number\n";
40     cin >> option;
41
42     switch (option)
43     {
44     case 1:
45         cin >> str;
46         result = isPalindrome(str);
47         Print(result);
48         break;
49     case 2:
50         cin >> num;
51         result = isPalindrome(num);
52         Print(result);
53         break;
54     default:
55         cerr << "Not correct option";
56         break;
57     }
58     return 0;
59 }

```

```

Enter option: 1 - for string , 2 - for number
2
12321
Is Palindrome

```

#### 4. algotester lab 2 variant 3

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.

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.

```

1  #include <iostream>
2  using namespace std;
3
4  int main ()
5  {
6      Click to collapse the range.
7      if (N<1 || N>1000)
8      {
9          cerr << "Error:Enter correct N";
10         return 0;
11     }
12
13     int arr[N];
14
15     for (int i = 0; i < N; i++)
16     {
17         cin >> arr[i];
18         if (arr[i]<1 || arr[i]>5)
19         {
20             cerr << "Error:Enter correct N";
21             return 0;
22         }
23     }
24     int left_idx = 0;
25     int right_idx = N-1;
26
27     while(true){
28         if (left_idx==right_idx)
29         {
30             cout << left_idx+1 <<" "<< right_idx+1 << endl;
31             cout << "Collision";
32             break;
33         }
34
35         if(left_idx>right_idx)
36         {
37             cout << left_idx+1 <<" "<< right_idx+1 << endl;
38             cout << "Miss";
39             break;
40         }
41

```

```

41
42         if(left_idx+1==right_idx)
43         {
44             cout << left_idx+1 <<" "<< right_idx+1 << endl;
45             cout << "Stopped";
46             break;
47         }
48
49         left_idx += arr[left_idx];
50         right_idx -= arr[right_idx];
51     }
52     return 0;
53 }

```

```

10
1 3 1 1 5 1 1 3 1 2
5 5
Collision

```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
3 days ago	C++ 23	Accepted	0.003	1.344	<a href="#">View</a>

Showing 1 to 1 of 1 rows

## 5. algotester lab 3 variant 2

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.

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int M;
6      int counter_both = 0;
7      int N;
8      cin >> N;
9      if (N<0 || N>100)
10     {
11         cerr << "Too large";
12         return 0;
13     }
14
15     int first_arr[N];
16     for (int i = 0; i < N; i++)
17     {
18         cin >> first_arr[i];
19         if (first_arr[i]<0 || first_arr[i]>100)
20         {
21             cerr << "Too large";
22             return 0;
23         }
24     }
25
26     cin >> M;
27     int second_arr[M];
28     for (int i = 0; i < M; i++)
29     {
30         cin >> second_arr[i];
31         if (second_arr[i]<0 || second_arr[i]>100)
32         {
33             cerr << "Too large";
34             return 0;
35         }
36     }
37
38     for (int i = 0; i < N; i++)
39     {
40         for (int j = 0; j < M; j++)
41         {
42             if (first_arr[i]==second_arr[j])
43             {
44                 counter_both++;
45                 break;
46             }
47         }
48     }
49     int counter_uni = (M+N) - counter_both;
50     cout << counter_both << '\n' << counter_uni;
51
52     return 0;
53 }
```

```
5
1 2 3 4 5
5
4 5 6 7 8
2
8
PS C:\Users\Maks\Documents\ai_programming_playground_2024\ai_11\maksym_martyn\epic_4> |
```



Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
2 days ago	C++ 23	Accepted	0.003	1.441	<a href="#">View</a>

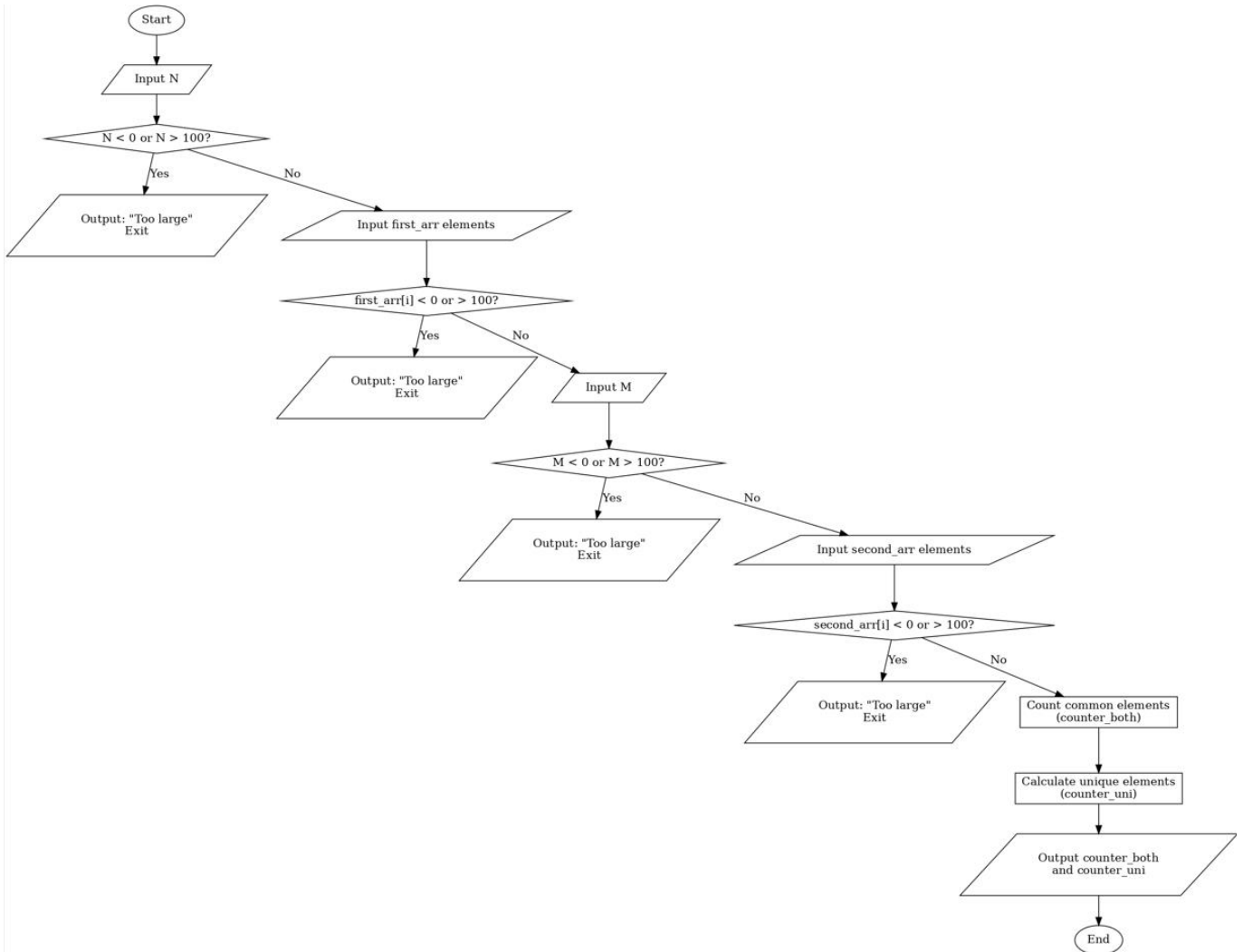
Showing 1 to 1 of 1 rows

## 6. Ввести двовимірний масив і знайти координати найменшого числа в ньому

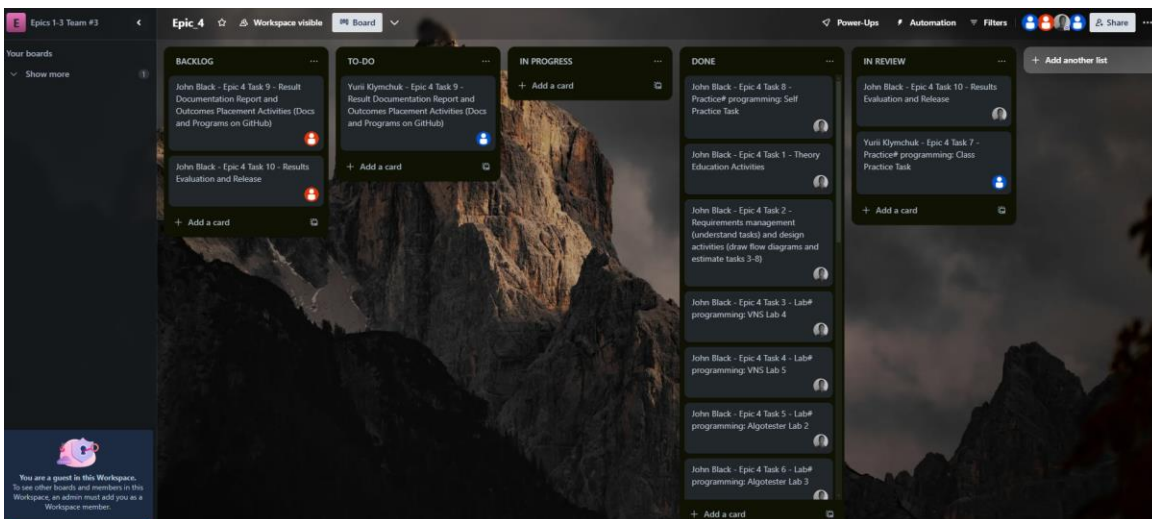
```

1  #include <iostream>
2  using namespace std;
3  int main ()
4  {
5      int rows, columns;
6      int** matrix=nullptr;
7      int user_num;
8      cout<<"Enter number of rows and columns ";
9      cin>>rows>>columns;
10     cout<<'\\n';
11
12     matrix=new int*[rows];
13     for (size_t i = 0; i < rows; i++)
14     {
15         matrix[i]=new int[columns];
16     }
17
18     for (size_t i = 0; i < rows; i++)
19     {
20         for (size_t j = 0; j < columns; j++)
21         {
22             cout<<'<<i+1<<"["<<j+1<<" = ";
23             cin>>matrix[i][j];
24         }
25     }
26     int min_element = matrix[0][0];
27     int idx_min_row = 0;
28     int idx_min_column = 0;
29
30     for (size_t i = 0; i < rows; i++)
31     {
32         for (size_t j = 0; j < columns; j++)
33         {
34             cout<<matrix[i][j]<<'\\t';
35         }
36         cout<<'\\n';
37     }
38
39     for (size_t i = 0; i < rows; i++)
40     {
41         for (size_t j = 0; j < columns; j++)
42         {
43             if (min_element>matrix[i][j])
44             {
45                 idx_min_row = i;
46                 idx_min_column=j;
47             }
48         }
49     }
50
51     cout<<"coordinates of min element is : "<<idx_min_row+1<<' ' <<idx_min_column+1;
52
53     for (size_t i = 0; i < rows; i++)
54     {
55         delete[] matrix[i];
56     }
57     delete[] matrix;
58     return 0;
59 }
```

## Блок схема до lab 3 var 2 algotester:



## Робота з командою



## Висновок:

Під час виконання лабораторної роботи я ознайомився з принципами роботи з одновимірними та двовимірними масивами, вказівниками, посиланнями, динамічними масивами та структурами даних у мові програмування C++.

Було реалізовано алгоритми обробки масивів і структур для вирішення завдань зі збереження, організації та аналізу даних.