

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



## Звіт

### про виконання лабораторних та практичних робіт блоку № 4

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

**Виконав:**  
Студент групи ІІІ-11  
Станько Олег Ігорович

Львів 2024

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

Мета: навчитися працювати з масивами, вказівниками, структурами.

### Теоретичні відомості

1. Класи пам'яті у C++
2. Вступ до Масивів і Вказівників
3. Одновимірні Масиви
4. Вказівники та Посилання
5. Двовимірні Масиви
6. Динамічні Масиви
7. Структури Даних
8. Вкладені Структури:
9. Використання структур
10. Алгоритми обробки та робота з масивами та структурами:

### Індивідуальний план опрацювання теорії

1. Класи пам'яті у C++  
[Динамічне виділення пам'яті в C++ / Уроки по C++ / aCode](#)  
[C++. Динамічне та статичне виділення пам'яті. Переваги та недоліки. Виділення пам'яті для одиночних змінних операторами new і delete. Можливі критичні ситуації при виділенні пам'яті. Ініціалізація при виділенні пам'яті | BestProg](#)  
Ознайомився з Статичною пам'яттю, динамічною пам'яттю, поняттям стеку, виділенням та вивільненням пам'яті.  
Витрачено 1 годину
2. Вступ до Масивів і Вказівників  
[C++. Масиви. Частина 1. Визначення масиву. Одновимірні масиви. Ініціалізація масиву | BestProg](#)  
[Масиви в C++ / Уроки по C++ / aCode](#)  
Ознайомився з масивами.  
Витрачено 30 хвилин.
3. Одновимірні Масиви  
[Поняття масиву. Одновимірні масиви](#)  
Ознайомився з Одновимірними Масивами  
Витрачено 20 хвилин
4. Вказівники та Посилання  
[Вказівники і масиви у C++ / aCode](#)  
Ознайомився з вказівником.  
Витрачено 30 хвилин.
5. Двовимірні Масиви  
[Двовимірні масиви в C++ / aCode](#)

Ознайомився з тим як оголошуються двовимірні масиви

Витрачено 30 хвилин.

6. Динамічні Масиви

[Динамічні масиви в C++ / Уроки по C++ / aCode](#)

Ознайомився з тим як оголошуються динамічні масиви

Витрачено 40 хвилин.

7. Структури Даних

[Структури в C++ / Уроки по C++ / aCode](#)

Ознайомився з тим як оголошуються структури.

Витрачено 40 хвилин.

8. Вкладені Структури

[C Structures - GeeksforGeeks](#)

ознайомився з вкладеними структурами.

Витрачено 40 хвилин.

9. Використання структури

[Overloading stream insertion \(<>\) operators in C++ - GeeksforGeeks](#)

ознайомився з введенням/виведенням структур

Витрачено 20 хвилин.

10. Алгоритми обробки та робота з масивами та структурами:

[Data Structures Tutorial - GeeksforGeeks](#)

ознайомився з алгоритмами обробки.

Витрачено 40 хвилин.

**Виконання роботи:**

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

**VNS LAB 4**

1) Реалізувати з використанням масиву однонаправлене кільце (перегляд можливий зліва направо, від останнього елемента можна перейти до першого).

2) Роздрукувати отриманий масив, починаючи з K-ого елемента і до K-1.

3) Додати в кільце перший і останній елементи.

4) Знищити з кільця парні елементи.

5) Роздрукувати отриманий масив, починаючи з K-ого елемента і до K-1

**VNS LAB 5**

Задано двовимірний масив  $N \times N$ . Послідовно розглядаються квадратні підмасиви, правий верхній елемент яких лежить на бічній діагоналі. У кожному такому підмасиві перебуває максимальний елемент. Шляхом перестановок рядків і стовпців (повністю) елемент треба перемістити в правий верхній кут підмасиву. Перевірити чи вийшла на бічній діагоналі спадаюча послідовність елементів.

**Algo lab 2**

У вас є масив  $r$  розміром  $N$ . Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей масив у масив сум, розміром  $N_{\text{new}}-1$  (розмір нового масиву після видалення елементів), який буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

### Algo lab 3

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.

### Practice

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.

### Вимоги:

1. Визначення функції:
  - a. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
  - a. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
  - a. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
  - b. *bool isPalindrome(ціле число);*
4. Рекурсія:
  - a. 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.

### Кроки реалізації

- Визначте та реалізуйте рекурсивну функцію *isPalindrome* для рядків.
- Визначте та реалізуйте перевантажену функцію *isPalindrome* для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.

## Self-practice task

Марічка казала, що в суботу піде разом із Зеником на олімпіаду з програмування. Зеник прийшов, а Марічки нема, підманула, підвела!

Замість олімпіади Марічка вирішила піти в місцевий парк атракціонів. Для того, щоб відвідати якнайбільшу кількість атракціонів, Марічка розпочне з найдешевшого атракціону, потім покатається на другому найдешевшому і так далі, доки в неї вистачить грошей. Зауважте, що Марічка не відвідує атракціон більше одного разу.

Відомо, що Марічка має  $k$  гривень, а в парку є  $n$  атракціонів. Для кожного атракціону відома його ціна  $c_j$ . Вам необхідно визначити кількість атракціонів, на яких покатається Марічка.

## Вхідні дані

У першому рядку задано два цілих числа  $k$  та  $n$  — кількість гривень та кількість атракціонів відповідно.

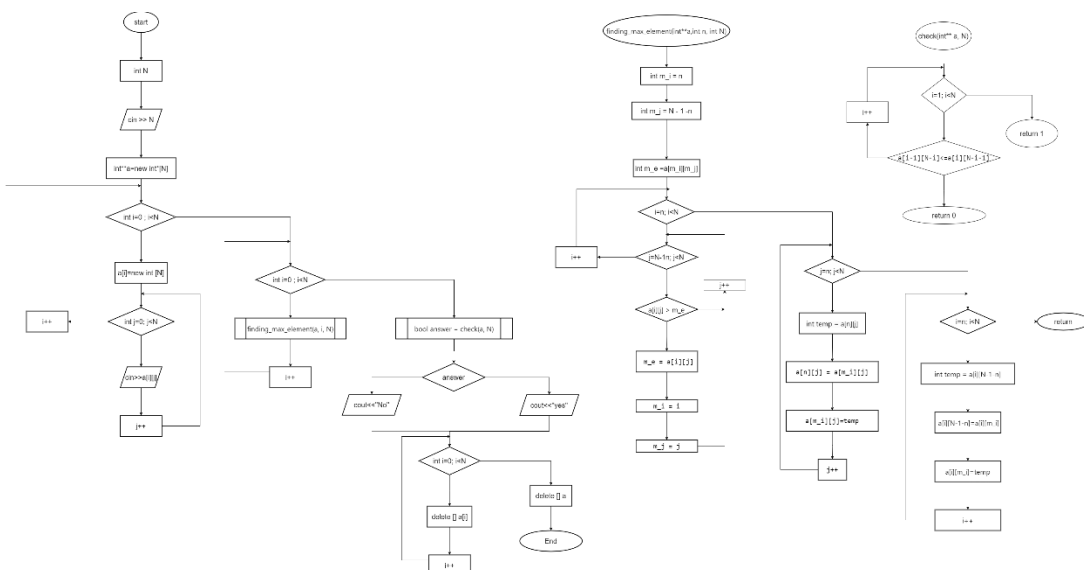
У другому рядку задано  $n$  цілих чисел  $c_j$  — ціна  $j$ -го атракціону.

## Вихідні дані

У єдиному рядку виведіть одне ціле число — кількість атракціонів, на яких покатається Марічка.

## 2. Блок-схеми

### VNS LAB 5



Коди:

## VNS LAB 4

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n = 100;
8      int a[n];
9      int size;
10     cout << "enter size of array: ";
11     cin >> size;
12     int k;
13     cout << "enter by which position you want to rotate: ";
14     cin >> k;
15     cout << "enter numbers of array: ";
16     for (int i = 0; i < size; i++)
17     {
18         cin >> a[i];
19     }
20     for (int i = k; i < k+size; i++)
21     {
22
23         cout << a[i%size] << " ";
24     }
25     int first;
26     cout << "\nenter new first number of array: ";
27     cin >> first;
28     cout << "enter new last number of array: ";
29     int last;
30     cin >> last;
31     for (int i = size; i > 0; i--)
32     {
33         a[i] = a[i - 1];
34     }
35     a[0] = first;
36     a[size+1]=last;
37     int b[size +1];
38     int m=0;
39     for (int i = 0; i < size+2; i++)
40     {
41         if (a[i]%2!=0)
42         {
43             b[m]=a[i];
44             m++;
45         }
46     }
47     cout << "Array after removing even elements: ";
48     for (int i = k; i < k+m; i++)
49     {
50         cout << b[i%m] << " ";
51     }
52 }
```

## VNS lab 5

```
al_programming_playground_2024 > al_11 > olen_stanko > epic_4 > C:\vns_lab_5_task_1\variant_18_olen_stanko\src\main.cpp
1  #include <iostream>
2  using namespace std;
3
4  void finding_max_element(int** a, int n, int N);
5  bool check(int** a, int N);
6
7  int main()
8  {
9      int N;
10     cin >> N;
11     int** a = new int*[N];
12     for (int i = 0; i < N; i++)
13     {
14         a[i] = new int[N];
15         for (int j = 0; j < N; j++)
16         {
17             cin >> a[i][j];
18         }
19     }
20
21     for (int i = 0; i < N; i++)
22     {
23         finding_max_element(a, i, N);
24     }
25
26     bool answer = check(a, N);
27     if (answer)
28     {
29         cout << "\nYes";
30     }
31     else
32     {
33         cout << "\nNo";
34     }
35
36     for (int i = 0; i < N; i++)
37     {
38         delete[] a[i];
39     }
40     delete[] a;
41
42     return 0;
43 }
44
45 void finding_max_element(int** a, int n, int N)
46 {
47     int m_i = n;
48     int m_j = N - 1 - n;
49     int m_e = a[m_i][m_j];
50
51     for (int i = n; i < N; i++)
52     {
53         for (int j = N - 1 - n; j < N; j++)
54         {
55             if (a[i][j] > m_e)
56             {
57                 m_e = a[i][j];
58                 m_i = i;
59                 m_j = j;
60             }
61         }
62     }
63
64     for (int j = 0; j < N; j++)
65     {
66         int temp = a[n][j];
67         a[n][j] = a[m_i][j];
68         a[m_i][j] = temp;
69     }
70
71     for (int i = 0; i < N; i++)
72     {
73         int temp = a[i][N - 1 - n];
74         a[i][N - 1 - n] = a[i][m_j];
75         a[i][m_j] = temp;
76     }
77 }
78
79 bool check(int** a, int N)
80 {
81     for (int i = 1; i < N; i++)
82     {
83         if (a[i-1][N-1] <= a[i][N-1])
84         {
85             return false;
86         }
87     }
88     return true;
89 }
```

## Algo lab 2

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int N;
8      cin>>N;
9      int r[N];
10     for (int i=0; i<N; i++)
11     {
12         cin>>r[i];
13     }
14     int a,b,c;
15     cin>>a>>b>>c;
16     int M=0;
17     int r1[N];
18     for (int i=0; i<N; i++)
19     {
20         if (r[i] != a && r[i] != b && r[i] != c)
21         {
22             r1[M] = r[i];
23             M++;
24         }
25     }
26     if (M>0)
27     {
28         M--;
29     }
30     cout<<M<<"\n";
31     for (int i=0; i<M; i++)
32     {
33         cout<<r1[i]+r1[i+1]<<" ";
34     }
35
36     return 0;
37 }
38
39
```

## Algo lab 3

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int N;
8      cin >> N;
9      int a[N];
10     for (int i = 0; i<N;i++)
11     {
12         cin >> a[i];
13     }
14     int M;
15     cin >> M;
16     int b[M];
17     for (int i = 0; i<M;i++)
18     {
19         cin >> b[i];
20     }
21     int s=0;
22     for (int i = 0; i<N;i++)
23     {
24         for (int j = 0; j<M; j++)
25         {
26             if (a[i]==b[j])
27             {
28                 s++;
29             }
30         }
31     }
32     cout<<s<<"\n";
33     cout<<N+M-s;
34     return 0;
35 }
36
```



## Practice

```
1  #include <iostream>
2
3  using namespace std;
4
5  bool isPalindrome(int num);
6
7  bool isPalindrome(const string& str, int start, int end);
8
9  int main()
10 {
11     int ans;
12     cout << "for exit type 0, for string type 1, for integer type 2\n";
13     cin >> ans;
14     while (ans!=0)
15     {
16         if (ans == 1)
17         {
18             string s;
19             cout << "Input a string: \n";
20             cin.ignore();
21             getline(cin, s);
22             bool res = isPalindrome(s, 0, s.length() - 1);
23             if (res)
24             {
25                 cout << "This string is palindrome\n";
26             }
27             else
28             {
29                 cout << "This string is not palindrome\n";
30             }
31         }
32         else if (ans == 2)
33         {
34             cout << "Input a number: \n";
35             int number;
36             cin >> number;
37             if (isPalindrome(number))
38             {
39                 cout << "This number is palindrome\n";
40             }
41             else
42             {
43                 cout << "This number is not palindrome\n";
44             }
45         }
46
47         cout << "for exit type 0, for string type 1, for integer type 2\n";
48         cin >> ans;
49     }
50 }
51
52
53 bool isPalindrome(const string& str, int start, int end)
54 {
55     if (start >= end)
56     {
57         return 1;
58     }
59     if (str[start] != str[end])
60     {
61         return 0;
62     }
63     return isPalindrome(str, start+1, end-1);
64 }
65
66
67 bool isPalindrome(int num)
68 {
69     if (num<0)
70     {
71         return 0;
72     }
73     int original = num;
74     int reversed = 0;
75     int digit;
76
77     while (num != 0) {
78         digit = num % 10;
79         reversed = reversed * 10 + digit;
80         num /= 10;
81     }
82     return original == reversed;
83 }
```

## Self practice work

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int k,n;
8      cin >> k >> n;
9      int c[n];
10     for (int i; i<n; i++)
11     {
12         cin >> c[i];
13     }
14     for (int i=0; i<n; i++)
15     {
16         bool is_change = false;
17         for (int j=0; j<n-1; j++)
18         {
19             if (c[j]>c[j+1])
20             {
21                 int temp = c[j];
22                 c[j] = c[j+1];
23                 c[j+1] = temp;
24                 is_change = true;
25             }
26         }
27         if (!is_change)
28         {
29             break;
30         }
31     }
32     int s = 0;
33     for (int i=0; i<n; i++)
34     {
35         if (k-c[i]>=0)
36         {
37             k-=c[i];
38             s++;
39         }
40     }
41     cout << s;
42 }
```

## Робота з командою

На жаль, на момент написання цього звіту ми ще не збиралися в зумі.

## Висновок

Завдяки цьому епіку я освоїв масиви і структури.