

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.»

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-13

Колбасюк Данило Іванович

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

Мета: Розширити знання про масиви та структури даних, застосувати алгоритми обробки та роботи з масивами та структурами.

Теоретичні відомості:

- лекції, практичні
- вказівки до лабораторних робіт ВНС
- <https://www.programiz.com/cpp-programming>
- [geeksforgeeks.org](https://www.geeksforgeeks.org)
- [w3schools.com/cpp](https://www.w3schools.com/cpp)

Виконання роботи

Завдання №1 Перевірка чи слово або число є паліндромом (practice task)

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»).

Завдання №2 VNS Lab#4(Варіант 18)

- 1) Реалізувати з використанням масиву однонаправлене кільце (перегляд можливий зліва направо, від останнього елемента можна перейти до першого).
- 2) Роздрукувати отриманий масив, починаючи з К-ого елемента і до К-1.
- 3) Додати в кільце перший і останній елементи.
- 4) Знищити з кільця парні елементи.
- 5) Роздрукувати отриманий масив, починаючи з К-ого елемента і до К-1.

Завдання №3 VNS Lab#5(Варіант 18)

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.

Завдання №4 Algotester Lab 2v3

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

Завдання №5 Algotester Lab 3v3

Вам дана стрічка s .

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу

підряд у стрічці замінити її на букву + кількість входжень підряд.

Завдання №6 Algostester self-practise task

Найбільша зростаюча підпослідовність

Обмеження: 2 сек., 256 МБ

Вам задано послідовність із n цілих чисел a_i . Ваша задача — знайти довжину найбільшої зростаючої підпослідовності заданої послідовності.

Вхідні дані

У першому рядку задано ціле число n .

У наступному рядку задано n цілих чисел — послідовність a_i .

Вихідні дані

Виведіть довжину найбільшої зростаючої підпослідовності.

Код, дизайн та оцінка часу

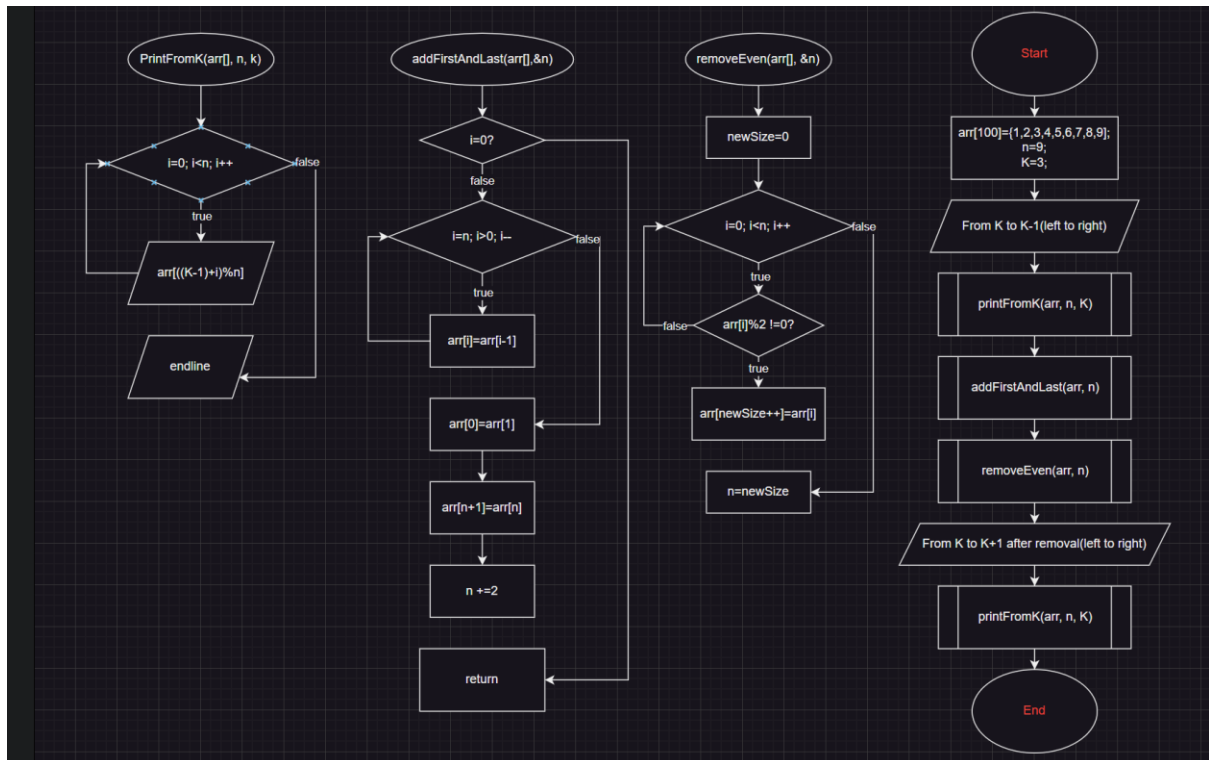
Завдання №1 Перевірка чи слово або число є паліндромом (practice task)

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  bool isPalindrome(const string &str, int start, int end)
6  {
7      if (start >= end)
8      {
9          return true;
10     }
11
12     if (str[start] != str[end])
13     {
14         return false;
15     }
16
17     return isPalindrome(str, start + 1, end - 1);
18 }
19
20 bool isPalindrome(int number)
21 {
22     int reversed = 0;
23     int n = number;
24     while (number > 0)
25     {
26         int x = number % 10;
27         reversed = reversed * 10 + x;
28         number /= 10;
29     }
30     return n == reversed;
31 }
32
33 int main()
34 {
35     int main()
36     {
37         string str;
38         cout << "Enter a word: ";
39         cin >> str;
40         cout << "This word is palindrome: " << isPalindrome(str, 0, str.size() - 1) << endl;
41
42         int n;
43         cout << "Enter a number: ";
44         cin >> n;
45         cout << "This number is palindrome: " << isPalindrome(n) << endl;
46
47         return 0;
48     }
49 }
```

```
Enter a word: 56765
This word is palindrome: 1
Enter a number: level
This number is palindrome: 1
```

Витрачено приблизно 2 години

Завдання №2 VNS Lab#4(Варіант 18)



```

1  #include <iostream>
2  using namespace std;
3
4  void printFromK(const int arr[], int n, int K) {
5      for (int i = 0; i < n; i++) {
6          cout << arr[((K-1) + i) % n] << " ";
7      }
8      cout << endl;
9  }
10
11 void addFirstAndLast(int arr[], int &n) {
12     if (n == 0){
13         return;
14     }
15     for (int i = n; i > 0; i--) {
16         arr[i] = arr[i - 1];
17     }
18     arr[0] = arr[1];
19     arr[n + 1] = arr[n];
20     n += 2;
21 }
22
23 void removeEven(int arr[], int &n) {
24     int newSize = 0;
25     for (int i = 0; i < n; i++) {
26         if (arr[i] % 2 != 0) {
27             arr[newSize++] = arr[i];
28         }
29     }
30     n = newSize;
31 }
32

```

```

23 √ void removeEven(int arr[], int &n) {
24     int newSize = 0;
25 √   for (int i = 0; i < n; i++) {
26 √       if (arr[i] % 2 != 0) {
27           arr[newSize++] = arr[i];
28       }
29     }
30     n = newSize;
31 }
32
33 √ int main() {
34     int arr[100] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
35     int n = 9;
36     int K = 3;
37
38     cout << endl << "From K to K-1 (left to right):" << endl;
39     printFromK(arr, n, K);
40
41     addFirstAndLast(arr, n);
42
43     removeEven(arr, n);
44     cout << "From K to K+1 after removal (left to right):" << endl;
45     printFromK(arr, n, K);
46
47     return 0;
48 }

```

From K to K-1 (left to right):

3 4 5 6 7 8 9 1 2

From K to K+1 after removal (left to right):

3 5 7 9 9 1 1

Витрачено приблизно 2.5 години

Завдання №3 VNS Lab#5(Варіант 18)

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4
5 const int N = 4;
6
7 void rearrangeMatrix(int matrix[N][N]);
8 bool isDiagonalDescending(const int matrix[N][N]);
9
10 int main() {
11     int matrix[N][N] = {
12         {12, 8, 21, 8},
13         {14, 5, 9, 10},
14         {2, 18, 7, 4},
15         {6, 11, 14, 26}
16     };
17
18     rearrangeMatrix(matrix);
19
20     cout << "Matrix after rearrangement:\n";
21     for (int i = 0; i < N; i++) {
22         for (int j = 0; j < N; j++) {
23             cout << matrix[i][j] << " ";
24         }
25         cout << endl;
26     }
27
28     if (isDiagonalDescending(matrix)) {
29         cout << "\nThe anti-diagonal forms a descending sequence.\n";
30     } else {
31         cout << "\nThe anti-diagonal does not form a descending sequence.\n";
32     }
33
34     return 0;
35 }
36
37 void rearrangeMatrix(int matrix[N][N]) {
```

Matrix after rearrangement:

6 11 14 26

12 8 21 8

2 18 7 4

14 5 9 10

The anti-diagonal forms a descending sequence.

```
37 void rearrangeMatrix(int matrix[N][N]) {
38     for (int i = 0; i < N; i++) {
39         int maxVal = matrix[i][N - 1 - i];
40         int maxRow = i, maxCol = N - 1 - i;
41
42         for (int r = i; r < N; r++) {
43             for (int c = i; c < N; c++) {
44                 if (matrix[r][c] > maxVal) {
45                     maxVal = matrix[r][c];
46                     maxRow = r;
47                     maxCol = c;
48                 }
49             }
50         }
51
52         if (maxRow != i) {
53             swap(matrix[i], matrix[maxRow]);
54         }
55
56         if (maxCol != N - 1 - i) {
57             for (int r = 0; r < N; r++) {
58                 swap(matrix[r][maxCol], matrix[r][N - 1 - i]);
59             }
60         }
61     }
62 }
63
64 bool isDiagonalDescending(const int matrix[N][N]) {
65     for (int i = 0; i < N - 1; i++) {
66         if (matrix[i][N - 1 - i] <= matrix[i + 1][N - 2 - i]) {
67             return false;
68         }
69     }
70     return true;
71 }
```

Витрачено приблизно 4 години

Завдання №4 Algotester Lab 2v3

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int n;
7      cout<<"Enter the amount of elements in array: ";
8      cin>>n;
9      vector<int>arr(n+1);
10     cout<<"Enter elements in array: ";
11     for (int i=1; i<=n; i++) {
12         cin>>arr[i];
13     }
14     int left_drone = 1;
15     int right_drone = n;
16
17     while (true)
18     {
19         if (left_drone == right_drone)
20         {
21             cout << left_drone<< " " << right_drone<< endl;
22             cout << "Collision" << endl;
23             break;
24         }
25         if (left_drone > right_drone)
26         {
27             cout << left_drone << " " << right_drone << endl;
28             cout << "Miss" << endl;
29             break;
30         }
31
32         if (left_drone == right_drone - 1)
33         {
34             cout << left_drone<< " " << right_drone << endl;
35             cout << "Stopped" << endl;
36             break;
37         }
38         left_drone = left_drone + arr[left_drone];
39         right_drone = right_drone - arr[right_drone];
40     }
41     return 0;
42 }
43 }
```

```
Enter the amount of elements in array: 10
Enter elements in array: 1 3 1 1 5 1 1 2 1 2
5 6
Stopped
Enter the amount of elements in array: 10
Enter elements in array: 1 3 1 1 5 1 1 3 1 2
5 5
Collision
Enter the amount of elements in array: 10
Enter elements in array: 1 3 1 1 5 1 1 5 1 2
5 3
Miss
```

Витрачено приблизно 45 хвилин

Завдання №5 Algotester Lab 3v3

```
#include <iostream>
#include <string>
using namespace std;

string compressString(const string &s) {
    string compressed = "";
    int count = 1;

    for (size_t i = 0; i < s.length(); ++i) {
        if (i + 1 < s.length() && s[i] == s[i + 1]) {
            count++;
        } else {
            compressed += s[i];
            if (count > 1) {
                compressed += to_string(count);
            }
            count = 1;
        }
    }
    return compressed;
}

int main() {
    string s;
    cout << "Enter string: ";
    cin >> s;

    string compressed = compressString(s);
    cout << "Compressed string: " << compressed << endl;

    return 0;
}
```

Enter string: AAAABBBBCQQQQ
Compressed string: A4B3CQ4

Enter string: AA
Compressed string: A2
Enter string: A
Compressed string: A

Витрачено приблизно 45 хвилин

Завдання №6 Algotester self-practise task

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> arr(n);
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];
    }
    vector<int> dp(n, 1);
    for (int i = 1; i < n; ++i) {
        for (int j = 0; j < i; ++j) {
            if (arr[j] < arr[i]) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
        }
    }
    cout << *max_element(dp.begin(), dp.end()) << endl;

    return 0;
}
```

0002 - Найбільша зростаюча підпоследовність

C++ 23

Зараховано

0.003

1.332

1898343

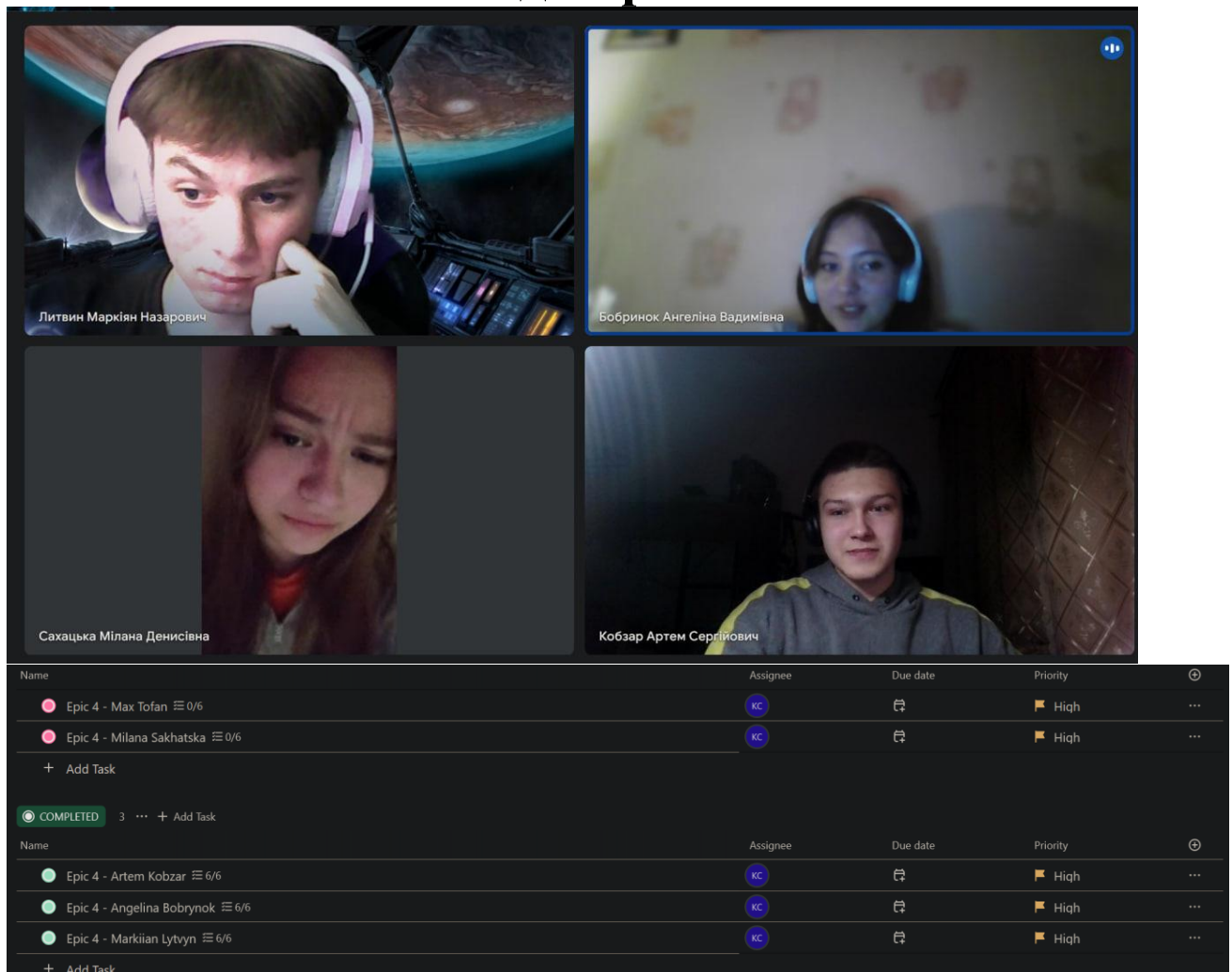
5

4 7 2 10 3

3

Витрачено приблизно 45 хвилин

Командна робота



The screenshot shows a video conference interface with four participants in a 2x2 grid. Below the video feeds is a task board with two sections: 'In Progress' and 'COMPLETED'.

Participants:

- Литвин Маркіян Назарович
- Бобринок Ангеліна Вадимівна
- Сахацька Мілана Денисівна
- Кобзар Артем Сергійович

Task Board:

Name	Assignee	Due date	Priority
Epic 4 - Max Tofan	КС		High
Epic 4 - Milana Sakhatska	КС		High
+ Add Task			
COMPLETED 3 + Add Task			
Epic 4 - Artem Kobzar	КС		High
Epic 4 - Angelina Bobrynok	КС		High
Epic 4 - Markiiian Lytvyn	КС		High
+ Add Task			

Висновок: у цьому блоці я застосував алгоритми обробки та роботи з масивами, а також розширив свої знання про масиви, структури, вказівники та посилання.