

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, 5

Лабораторної на Algotester №2, 3

Практичних Робіт до блоку № 4

Виконала:

Студентка групи ШІ-12

Олійник Божена

Львів 2024

Тема роботи

1. Використання одновимірних, двовимірних та динамічних масивів.
2. Використання структур даних, вкладених структур даних.
3. Використання вказівників та посилань.

Мета роботи

1. Навчитись створювати та використовувати одновимірні та двовимірні масиви, а також динамічні масиви.
2. Навчитись користуватись структурами даних.
3. Навчитись використовувати вказівники та посилання.

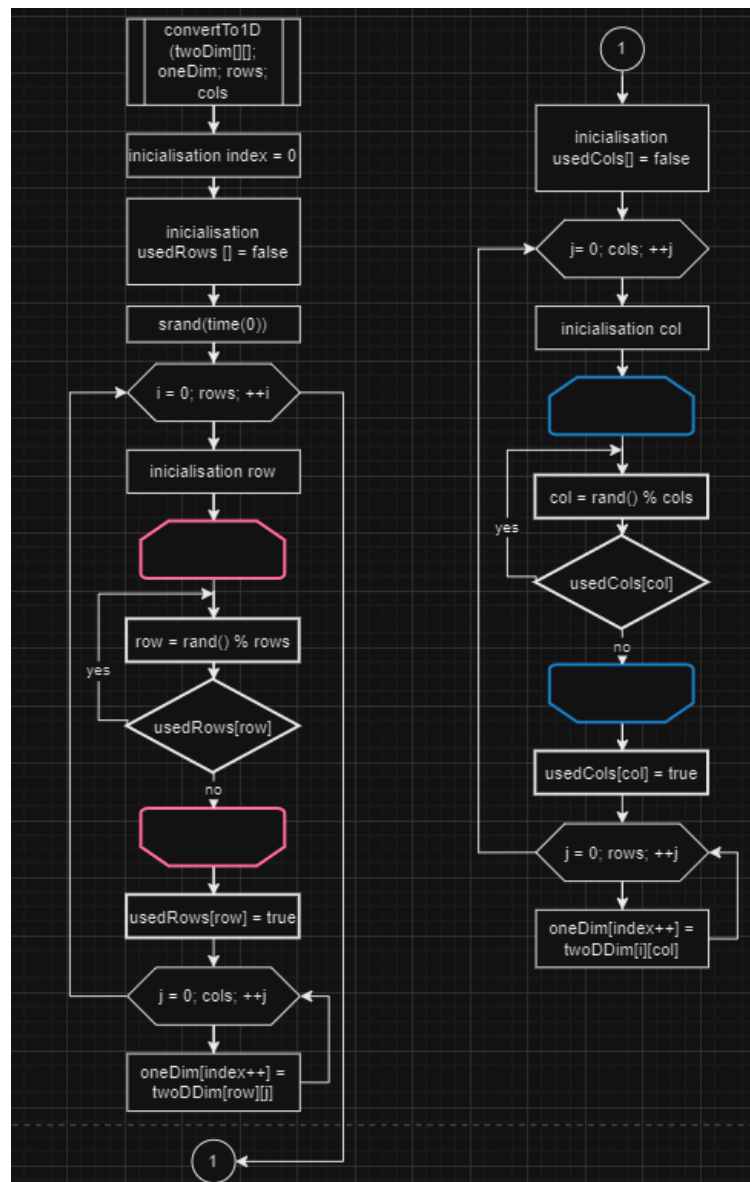
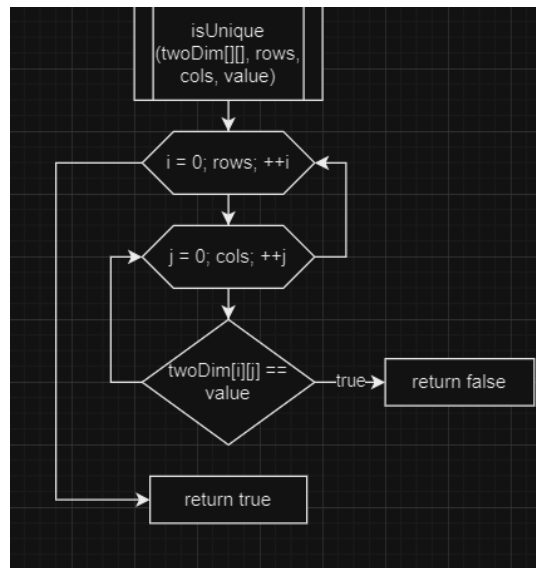
Теоретичні відомості

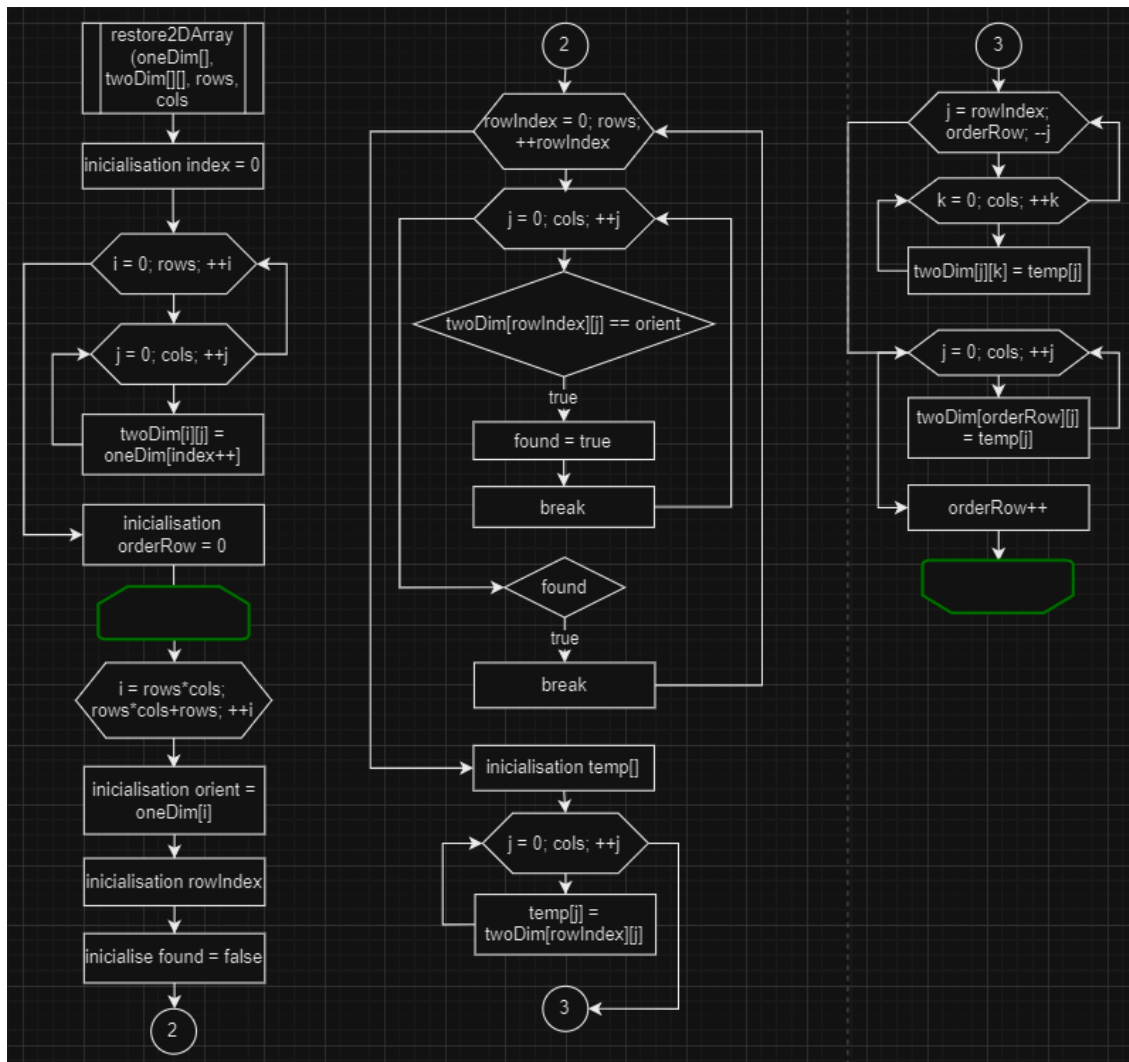
1. Масиви:
<https://acode.com.ua/urok-77-masyvy/>
<https://acode.com.ua/urok-81-bagatovymirni-masyvy/>
2. Структури:
<https://acode.com.ua/urok-64-struktury/#toc-7>
3. Вказівники:
<https://acode.com.ua/urok-84-vkazivnyky/#toc-2>
4. Посилання:
<https://acode.com.ua/urok-92-posylannya/#toc-0>

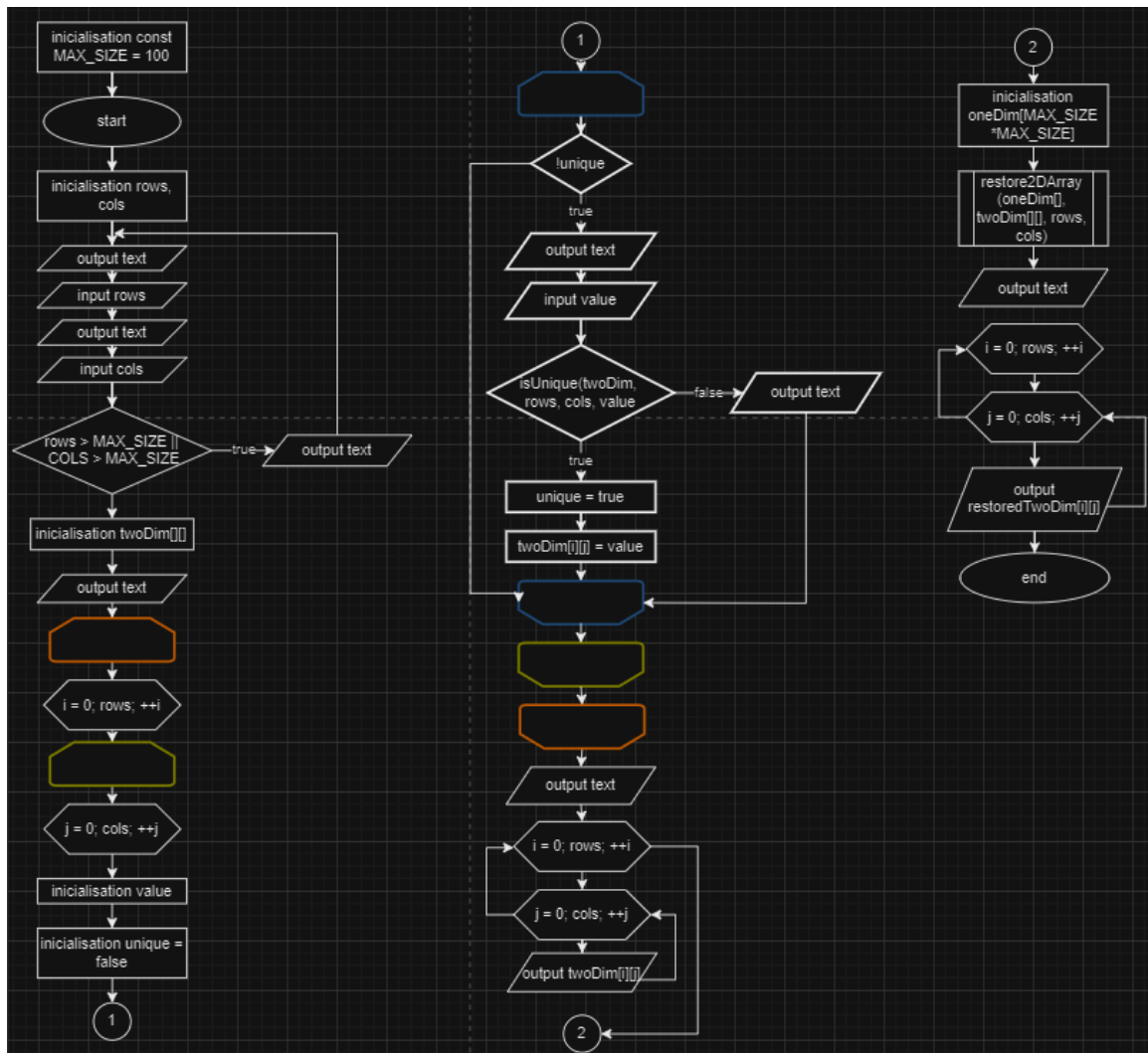
Виконання роботи

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7) (2 год)

VNS Lab 5







Task 3 – Lab# programming: VNS Lab 4 (30xв)

```

1  #include <iostream>
2  #include <algorithm>
3  using namespace std;
4
5  const int N = 100;
6
7  void printRing(int arr[], int length, int K) {
8      cout << "Елементи кільця з індексу " << K << " до " << (K + 1) % length << " (у зворотньому порядку):\n";
9      int index = K;
10     do {
11         cout << arr[index] << " ";
12         index = (index - 1 + length) % length;
13     } while (index != ((K + 1) % length));
14     cout << arr[index] << " ";
15     cout << endl;
16 }
17
18 int removeOdd(int arr[], int length) {
19     int newLength = 0;
20     for (int i = 0; i < length; ++i) {
21         if (arr[i] % 2 == 0) {
22             arr[newLength++] = arr[i];
23         }
24     }
25     return newLength;
26 }
27
28 int main() {
29     int a[N];
30     int reallength;
31
32     wrongLength:
33     cout << "Введіть реальну довжину масиву (не більше " << N << "): ";
34     cin >> reallength;
35     if (reallength > N || reallength <= 0) {
36         cout << "Невірна довжина масиву!" << endl;
37         goto wrongLength;
38         return 1;
39     }
40
41     for (int i = 0; i < reallength; ++i) {
42         cout << "Введіть " << i << " елемент масиву: ";
43         cin >> a[i];
44     }

```

```

47     int K;
48     wrongIndex:
49     cout << "Введіть індекс K (0 <= K < " << reallength << "): ";
50     cin >> K;
51     if (K < 0 || K >= reallength) {
52         cout << "Невірний індекс K!" << endl;
53         goto wrongIndex;
54         return 1;
55     }
56
57     printRing(a, reallength, K);
58
59     sort(a, a + reallength, greater<int>());
60
61     reallength = removeOdd(a, reallength);
62
63     cout << "Посортований масив без непарних елементів: " << endl;
64     for(int i = 0; i < reallength; i++){
65         cout << a[i] << " ";
66     }
67     cout << endl;
68
69     printRing(a, reallength, K);
70
71     return 0;
72 }
73

```

Task 4 - Lab# programming: VNS Lab 5 (2.5год)

```

1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  const int MAX_SIZE = 100;
7
8  bool isUnique(int twoDim[][MAX_SIZE], int rows, int cols, int value)
9  {
10     for (int i = 0; i < rows; ++i)
11     {
12         for (int j = 0; j < cols; ++j)
13         {
14             if (twoDim[i][j] == value)
15             {
16                 return false;
17             }
18         }
19     }
20     return true;
21 }
22
23 void convertTo1D(int twoDim[][MAX_SIZE], int oneDim[], int rows, int cols)
24 {
25     int index = 0;
26
27     bool usedRows[MAX_SIZE] = {false};
28     srand(time(0));
29
30     for (int i = 0; i < rows; ++i)
31     {
32         int row;
33         do
34         {
35             row = rand() % rows;
36         } while (usedRows[row]);
37         usedRows[row] = true;
38
39         for (int j = 0; j < cols; ++j)
40         {
41             oneDim[index++] = twoDim[row][j];
42         }
43     }
44     bool usedCols[MAX_SIZE] = {false};
45     for (int j = 0; j < cols; ++j)
46     {
47         int col;
48         do
49         {
50             col = rand() % cols;
51         } while (usedCols[col]);
52         usedCols[col] = true;
53
54         for (int i = 0; i < rows; ++i)
55         {
56             oneDim[index++] = twoDim[i][col];
57         }
58     }
59 }
60
61 void restore2DArray(int oneDim[], int twoDim[][MAX_SIZE], int rows, int cols)
62 {
63     int index = 0;
64
65     for (int i = 0; i < rows; ++i)
66     {
67         for (int j = 0; j < cols; ++j)
68         {
69             twoDim[i][j] = oneDim[index++];
70         }
71     }
72
73     int orderRow = 0;
74     for (int i = rows * cols; i < rows * cols + rows; ++i)
75     {
76         int orient = oneDim[i];
77
78         int rowIndex;
79         bool found = false;
80         for (rowIndex = 0; rowIndex < rows; ++rowIndex)
81         {
82             for (int j = 0; j < cols; ++j)
83             {
84                 if (twoDim[rowIndex][j] == orient)
85                 {
86                     found = true;
87                     break;
88                 }
89             }
90             if (found)
91                 break;
92         }
93
94         int temp[MAX_SIZE];
95         for (int j = 0; j < cols; ++j)
96         {
97             temp[j] = twoDim[rowIndex][j];
98         }
99
100         for (int j = rowIndex; j > orderRow; --j)
101         {
102             for (int k = 0; k < cols; ++k)
103             {
104                 twoDim[j][k] = twoDim[j - 1][k];
105             }
106         }
107
108         for (int j = 0; j < cols; ++j)
109         {
110             twoDim[orderRow][j] = temp[j];
111         }
112         orderRow++;
113     }
114 }
115
116 int main()
117 {
118     int rows, cols;
119
120     size:
121     cout << "Введіть кількість рядків: ";
122     cin >> rows;
123     cout << "Введіть кількість стовпців: ";
124     cin >> cols;
125
126     if (rows > MAX_SIZE || cols > MAX_SIZE)
127     {
128         cout << "Розмірність перевищує максимальний дозволений розмір!" << endl;
129         goto size;
130     }
131
132     int twoDim[MAX_SIZE][MAX_SIZE];
133     cout << "Введіть елементи двовимірного масиву:" << endl;
134     for (int i = 0; i < rows; ++i)
135     {
136         for (int j = 0; j < cols; ++j)
137         {
138             int value;
139             bool unique = false;
140             while (!unique)
141             {
142                 cout << "Елемент [" << i << "][" << j << "]: ";
143                 cin >> value;
144
145                 if (isUnique(twoDim, rows, cols, value))
146                 {
147                     unique = true;
148                     twoDim[i][j] = value;
149                 }
150                 else
151                 {
152                     cout << "Цей елемент уже існує в масиві, спробуйте ще раз." << endl;
153                 }
154             }
155         }
156     }
157
158     cout << "Двовимірний масив:" << endl;
159     for (int i = 0; i < rows; ++i)
160     {
161         for (int j = 0; j < cols; ++j)
162         {
163             cout << twoDim[i][j] << " ";
164         }
165         cout << endl;
166     }
167
168     int oneDim[MAX_SIZE * MAX_SIZE];
169     convertTo1D(twoDim, oneDim, rows, cols);
170
171     cout << "Одновимірний масив (після переписування):" << endl;
172     for (int i = 0; i < rows * cols * 2; ++i)
173     {
174         cout << oneDim[i] << " ";
175     }

```

```

170
171     cout << "Одновимірний масив (після переписування):" << endl;
172     for (int i = 0; i < rows * cols * 2; ++i)
173     {
174         cout << oneDim[i] << " ";
175     }
176     cout << endl;
177
178     int restoredTwoDim[MAX_SIZE][MAX_SIZE];
179     restore2DArray(oneDim, restoredTwoDim, rows, cols);
180
181     cout << "Відновлений двовимірний масив:" << endl;
182     for (int i = 0; i < rows; ++i)
183     {
184         for (int j = 0; j < cols; ++j)
185         {
186             cout << restoredTwoDim[i][j] << " ";
187         }
188         cout << endl;
189     }
190
191     return 0;
192 }
193

```

Task 5 - Lab# programming: Algotester Lab 2 (20хв)

```

2  #include <iostream>
3  #include <climits>
4  using namespace std;
5
6  int main()
7  {
8      int N, m;
9      cin >> N;
10
11     if (N <= 2) {
12         cout << 0;
13         return 0;
14     }
15
16     int* r = new int[N];
17
18     for (int i = 0; i < N; i++)
19     {
20         cin >> r[i];
21     }
22
23     int min = INT_MAX, max = INT_MIN;
24     bool doubleMin = false, doubleMax = false;
25
26     for (int i = 0; i < N; i++)
27     {
28         if (r[i] < min)
29         {
30             min = r[i];
31             doubleMin = false;
32         }
33         else if (r[i] == min)
34         {
35             doubleMin = true;
36         }
37
38         if (r[i] > max)
39         {
40             max = r[i];
41             doubleMax = false;
42         }
43         else if (r[i] == max)
44         {
45             doubleMax = true;
46         }
47     }
48
49     int nextMin = INT_MAX, nextMax = INT_MIN;
50     for (int i = 0; i < N; i++)
51     {
52         if (r[i] > min && r[i] < nextMin)
53         {
54             nextMin = r[i];
55         }
56         if (r[i] < max && r[i] > nextMax)
57         {
58             nextMax = r[i];
59         }
60     }
61
62     if (doubleMin)
63     {
64         nextMin = min;
65     }
66     if (doubleMax)
67     {
68         nextMax = max;
69     }
70
71     int diffMax = nextMax - min;
72     int diffMin = max - nextMin;
73
74     if (diffMin <= diffMax)
75     {
76         m = max - nextMin;
77     }
78     else
79     {
80         m = nextMax - min;
81     }
82
83     cout << m << endl;
84
85     delete[] r;
86     return 0;
87 }
88

```


Task 5 - Lab# programming: Algotester Lab 3 (20xв)

```
1  #include <iostream>
2  #include <algorithm>
3  using namespace std;
4
5  int main()
6  {
7      int N, M;
8      cin >> N;
9
10     int n[N];
11     for (int i = 0; i < N; i++)
12     {
13         cin >> n[i];
14     }
15
16     cin >> M;
17     int m[M];
18     for (int i = 0; i < M; i++)
19     {
20         cin >> m[i];
21     }
22
23     int combined[N + M];
24     for (int i = 0; i < N; i++)
25     {
26         combined[i] = n[i];
27     }
28     for (int i = 0; i < M; i++)
29     {
30         combined[N + i] = m[i];
31     }
32
33     sort(combined, combined + N + M);
34
35     int uniqueCount = 0;
36     for (int i = 0; i < N + M; i++)
37     {
38         if (i == 0 || combined[i] != combined[i - 1])
39         {
40             uniqueCount++;
41         }
42     }
43
44     int commonCount = 0;
45     int i = 0, j = 0;
46     sort(n, n + N);
47     sort(m, m + M);
48     while (i < N && j < M)
49     {
50         if (n[i] == m[j])
51         {
52             commonCount++;
53             int value = n[i];
54             while (i < N && n[i] == value) i++;
55             while (j < M && m[j] == value) j++;
56         }
57         else if (n[i] < m[j])
58         {
59             i++;
60         }
61         else
62         {
63             j++;
64         }
65     }
66
67     cout << commonCount << endl;
68     cout << uniqueCount << endl;
69
70     return 0;
71 }
72
```

Task 7 - Practice# programming: Class Practice Task (30xв)

```

1  #include <iostream>
2  #include <string>
3  #include <cmath>
4
5  using namespace std;
6
7  bool isPalindrome(const string &str, unsigned int start, unsigned int end)
8  {
9      if (start >= end)
10     {
11         return true;
12     }
13     if (str[start] != str[end])
14     {
15         return false;
16     }
17     return isPalindrome(str, start + 1, end - 1);
18 }
19
20 bool isPalindrome(int num)
21 {
22     if (num < 0)
23         return false;
24
25     if (num >= 0 && num <= 9)
26         return true;
27
28     int digits = static_cast<int>(log10(num)) + 1;
29     int divisor = 1;
30
31     for (int i = 0; i < digits - 1; ++i)
32     {
33         divisor *= 10;
34     }
35
36     while (num != 0)
37     {
38         int firstDigit = num / divisor;
39         int lastDigit = num % 10;
40
41         if (firstDigit != lastDigit)
42         {
43             return false;
44         }
45
46         num = (num % divisor) / 10;
47         divisor /= 100;
48     }
49
50     return true;
51 }
52
53 int main()
54 {
55     string word;
56     cout << "Enter a word: ";
57     cin >> word;
58
59     if (isPalindrome(word, 0, static_cast<unsigned int>(word.length()) - 1))
60     {
61         cout << "This word is a palindrome!" << endl;
62     }
63     else
64     {
65         cout << "This word is not a palindrome!" << endl;
66     }
67
68     int num;
69     cout << "Enter a number: ";
70     cin >> num;
71
72     if (isPalindrome(num))
73     {
74         cout << "This number is a palindrome!" << endl;
75     }
76     else
77     {
78         cout << "This number is not a palindrome!" << endl;
79     }
80
81     return 0;
82 }

```

Task 8 - Practice# programming: Self Practice Task (25xb)

```
1  #include <iostream>
2  #include <string>
3  #include <cmath>
4
5  using namespace std;
6
7  bool isPalindrome(const string &str, unsigned int start, unsigned int end)
8  {
9      if (start >= end)
10     {
11         return true;
12     }
13     if (str[start] != str[end])
14     {
15         return false;
16     }
17     return isPalindrome(str, start + 1, end - 1);
18 }
19
20 bool isPalindrome(int num)
21 {
22     if (num < 0)
23         return false;
24
25     if (num >= 0 && num <= 9)
26         return true;
27
28     int digits = static_cast<int>(log10(num)) + 1;
29     int divisor = 1;
30
31     for (int i = 0; i < digits - 1; ++i)
32     {
33         divisor *= 10;
34     }
35
36     while (num != 0)
37     {
38         int firstDigit = num / divisor;
39         int lastDigit = num % 10;
40
41         if (firstDigit != lastDigit)
42         {
43             return false;
44         }
45
46         num = (num % divisor) / 10;
47         divisor /= 100;
48     }
49
50     return true;
51 }
52
53 int main()
54 {
55     string word;
56     cout << "Enter a word: ";
57     cin >> word;
58
59     if (isPalindrome(word, 0, static_cast<unsigned int>(word.length()) - 1))
60     {
61         cout << "This word is a palindrome!" << endl;
62     }
63     else
64     {
65         cout << "This word is not a palindrome!" << endl;
66     }
67
68     int num;
69     cout << "Enter a number: ";
70     cin >> num;
71
72     if (isPalindrome(num))
73     {
74         cout << "This number is a palindrome!" << endl;
75     }
76     else
77     {
78         cout << "This number is not a palindrome!" << endl;
79     }
80
81     return 0;
82 }
83
```

Зустрічі з командою

З командою зустрічалися двічі, на зустрічах обговорювали питання та прогрес по епіку.



Висновок

В ході даного епіку я навчилась використовувати одновимірні та двовимірні масиви, а також створювати та користуватись динамічними масивами. Також я використовувала структури даних.