

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



Звіт

про виконання лабораторних та практичних робіт блоку № 4
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 ШІ-12
Горішний Микола Мар'янович

Львів 2024

Тема роботи:

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

Мета роботи:

Розібратись з класами пам'яті, зокрема зі статичною, динамічною пам'яттю, поняттям стеку, виділення та вивільнення пам'яті.

Дослідження одновимірних і двовимірних масивів для зберігання і впорядкування даних, що забезпечує швидкий доступ і обробку великих обсягів інформації.

Дослідження вказівників та посилань для розуміння адресації пам'яті та оптимізації використання ресурсів, що дозволяє ефективніше працювати з динамічними структурами даних.

Дослідження динамічних масивів для створення програм із змінною кількістю елементів, що підвищує гнучкість і адаптивність коду.

Дослідження структур даних та вкладених структур для організації складних об'єктів, що забезпечує кращу структуру і читабельність програмного коду.

Дослідження алгоритмів обробки масивів і структур для реалізації ефективної обробки даних, що сприяє написанню оптимізованих і масштабованих програм.

Теоретичні відомості:

У цій роботі розглядаються основні принципи роботи з масивами та структурами даних, зокрема одновимірні й двовимірні масиви для організації і зберігання великих обсягів даних. Особливу увагу приділено вказівникам і посиланням як засобам управління пам'яттю та ефективного доступу до даних. Розглянуто динамічні масиви, які забезпечують гнучке управління розміром даних під час виконання програми. Досліджено основи структур даних і вкладених структур для створення складних, логічно організованих об'єктів. Описано алгоритми обробки масивів і структур, що дозволяють ефективно виконувати операції пошуку, сортування і модифікації даних, покращуючи оптимізацію коду.

Джерела:

- 1) <https://www.youtube.com/watch?v=V2g3B9Zbh4Q&t=1s>
- 2) <https://www.youtube.com/watch?v=s7H2ScLcH88;>
- 3) [https://acode.com.ua/urok-81-bagatovymirni-masyvy/;](https://acode.com.ua/urok-81-bagatovymirni-masyvy/)
- 4) <https://www.youtube.com/watch?v=qUX4xCRB8FM&t=445s;>

Lab# programming: VNS Lab 4

3.

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити всі елементи, які дорівнюють 0.
- 4) Додати після першого парного елемента масиву елемент зі значенням $M[I-1] + 2$.
- 5) Роздрукувати отриманий масив.

ic_4_Lab_4_Random_Number (Global Scope)

```
1  #include <iostream>
2  #include <vector>
3  #include <cstdlib>
4  #include <ctime>
5  #include <algorithm>
6  using namespace std;
7
8  int main()
9  {
10     srand(time(0));
11
12     int n = 10;
13     vector<int> arr(n);
14
15     for (int& num : arr)
16     {
17         num = rand() % 11 - 5;
18     }
19
20     cout << "Start massive : ";
21     for (int num : arr)
22     {
23         cout << num << " ";
24     }
25     cout << endl;
26
27     arr.erase(remove(arr.begin(), arr.end(), 0), arr.end());
28
29     for (size_t i = 0; i < arr.size(); ++i)
30     {
31         if (arr[i] % 2 == 0)
32         {
33             arr.insert(arr.begin() + i + 1, arr[i - 1] + 2);
34             break;
35         }
36     }
37
38 }
```

```
39     }
40
41     cout << "New massive ";
42     for (int num : arr)
43     {
44         cout << num << " ";
45     }
46     cout << endl;
47
48
49
50
51
52     return 0;
53 }
```

Lab# programming: VNS Lab 5

3. Написати процедуру для підсумовування матриць. З її допомогою скласти вихідну матрицю й транспоновану (тобто отриману поворотом вихідної на 90°).

Epic_4_Lab_5_Matrix (Global Scope)

```
1  #include <iostream>
2  using namespace std;
3
4
5  void sumMatrices(int mat1[3][3], int mat2[3][3], int result[3][3])
6  {
7      for (int i = 0; i < 3; ++i)
8      {
9          for (int j = 0; j < 3; ++j)
10         {
11             result[i][j] = mat1[i][j] + mat2[i][j];
12         }
13     }
14 }
15
16
17 void transposeMatrix(int mat[3][3], int result[3][3])
18 {
19     for (int i = 0; i < 3; ++i)
20     {
21         for (int j = 0; j < 3; ++j)
22         {
23             result[j][i] = mat[i][j];
24         }
25     }
26 }
27
28
29
30 void printMatrix(int mat[3][3])
31 {
32     for (int i = 0; i < 3; ++i)
33     {
34         for (int j = 0; j < 3; ++j)
35         {
36             cout << mat[i][j] << " ";
37         }
38         cout << endl;
39     }
40 }
```

```

40     }
41 }
42
43 int main()
44 {
45     int mat1[3][3], mat2[3][3];
46     int sumResult[3][3];
47     int transposeResult[3][3];
48
49     cout << " Enter elemements for first matrix 3x3:" << endl;
50     for (int i = 0; i < 3; ++i)
51     {
52         cout << "Row : " ;
53         for (int j = 0; j < 3; ++j)
54         {
55             cin >> mat1[i][j];
56         }
57     }
58
59     cout << " Enter elemements for second matrix 3x3:" << endl;
60     for (int i = 0; i < 3; ++i)
61     {
62         cout << "Row : " ;
63         for (int j = 0; j < 3; ++j)
64         {
65             cin >> mat2[i][j];
66         }
67     }
68
69
70     sumMatrices(mat1, mat2, sumResult);
71     cout << "Result for sum of two matrix:" << endl;
72     printMatrix(sumResult);
73
74
75     transposeMatrix(sumResult, transposeResult);
76     cout << "Transposed matrix :" << endl;

```

```

sumMatrices(mat1, mat2, sumResult);
cout << "Result for sum of two matrix:" << endl;
printMatrix(sumResult);

transposeMatrix(sumResult, transposeResult);
cout << "Transposed matrix :" << endl;
printMatrix(transposeResult);

return 0;
}

```

Lab# programming: Algotester Lab 2

```
Epic_4_Algo_Lab_2 (Global Scope)
1  #include <iostream>
2  #include <unordered_set>
3  using namespace std;
4
5
6  int main()
7  {
8      int N;
9      cin >> N;
10
11     int* r = new int[N];
12     for (int i = 0; i < N; ++i)
13     {
14         cin >> r[i];
15     }
16     int a, b, c;
17     cin >> a >> b >> c;
18
19     unordered_set<int> to_remove = { a, b, c }; // щоб зберігати видалені числа
20
21     int* NewMassive = new int[N];
22     int NewMassive_size = 0;
23
24     for (int i = 0; i < N; ++i) {
25         if (to_remove.find(r[i]) == to_remove.end())
26         {
27             NewMassive[NewMassive_size++] = r[i];
28         }
29     }
30
31     if (NewMassive_size < 2)
32     {
33         cout << "0" << endl;
34         cout << "" << endl;
35     }
36     else {
37         int* sums = new int[NewMassive_size - 1];
38         for (int i = 0; i < NewMassive_size - 1; ++i)
39         {
40             sums[i] = NewMassive[i] + NewMassive[i + 1];
41         }
42
43
44         cout << NewMassive_size - 1 << endl;
45         for (int i = 0; i < NewMassive_size - 1; ++i)
46         {
47             cout << sums[i] << " ";
48         }
49         cout << endl;
50
51         delete[] sums;
52     }
53
54     delete[] r;
55     delete[] NewMassive;
56
57
58     return 0;
59 }
60
```


Lab# programming: Algotester Lab 3

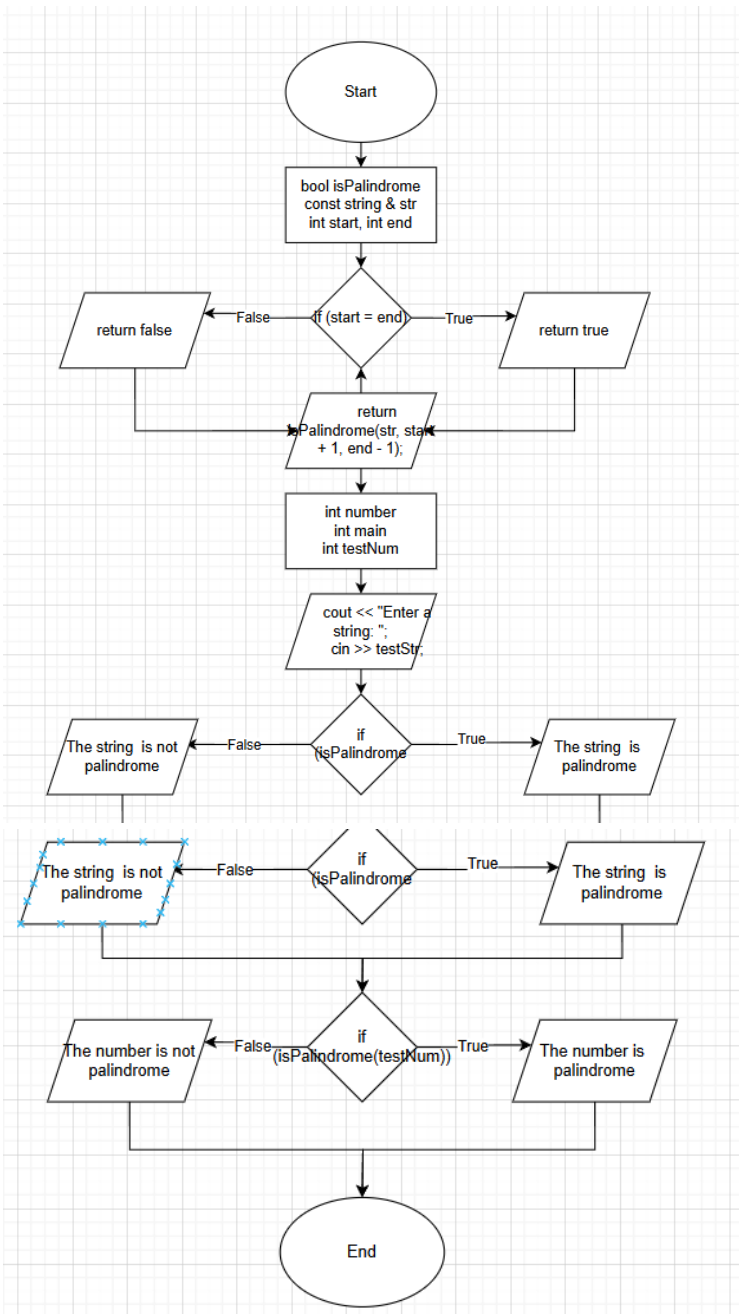
```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string s;
8      cin >> s;
9
10     string result = "";
11     int n = s.length();
12
13     int i = 0;
14     while (i < n)
15     {
16         char currentChar = s[i];
17         int count = 1;
18
19         while (i + 1 < n && s[i + 1] == currentChar)
20         {
21             count++;
22             i++;
23         }
24
25         if (count > 1)
26         {
27             result += currentChar + to_string(count);
28         }
29         else {
30             result += currentChar;
31         }
32
33         i++;
34     }
35
36     cout << result << endl;
37
38
39     return 0;
40 }
```

Practice# programming: Class 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»).

```
Epic_4_Practice_Level (Global Scope)
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5
6
7  bool isPalindrome(const string & str, int start, 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
21 bool isPalindrome(int number)
22 {
23     string str = to_string(number);
24     return isPalindrome(str, 0, str.length() - 1);
25 }
26
27 int main()
28 {
29     string testStr;
30     cout << "Enter a string: ";
31     cin >> testStr;
32     if (isPalindrome(testStr, 0, testStr.length() - 1))
33     {
34         cout << "The string " << testStr << " is a palindrome." << endl;
35     }
36     else
37     {
38         cout << "The string " << testStr << " is not a palindrome." << endl;
39     }
40
41     int testNum;
42     cout << "Enter a number: ";
43     cin >> testNum;
44     if (isPalindrome(testNum))
45     {
46         cout << "The number" << testNum << " is a palindrome." << endl;
47     }
48     else
49     {
50         cout << "The number " << testNum << " is not palindrome." << endl;
51     }
52
53
54     return 0;
55 }
56
57
58
```

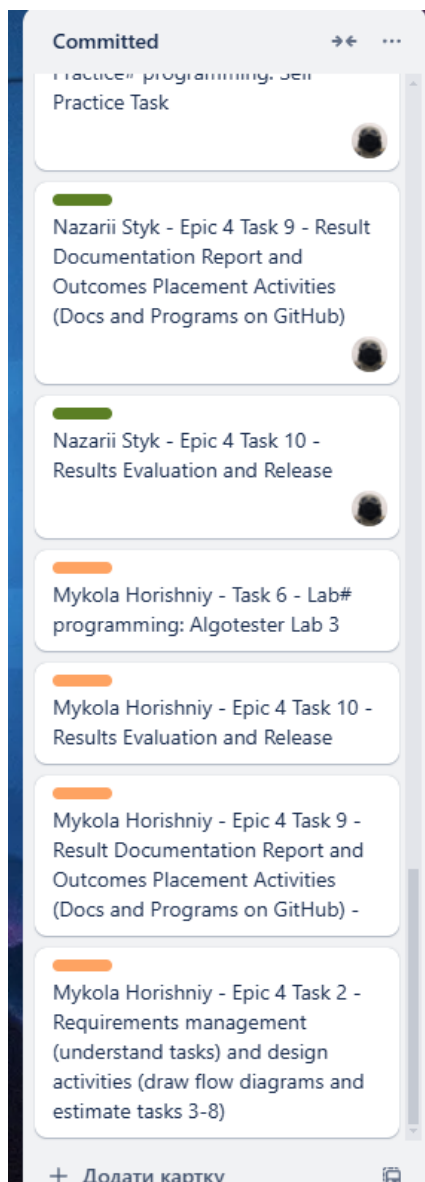
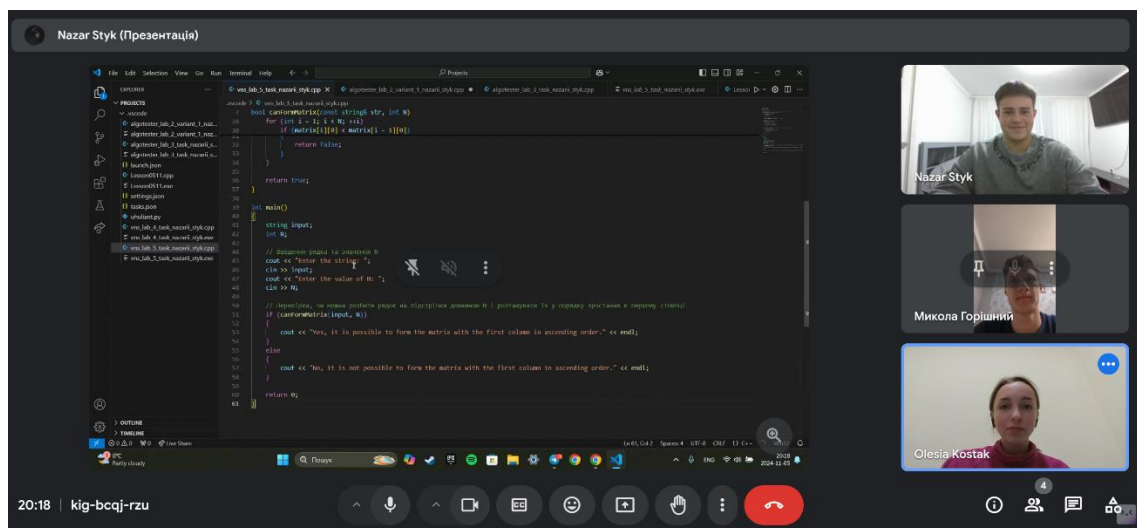


Practice# programming: Self Practice Task

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int n;
7     cin >> n;
8     int numbers[] = {500, 200, 100, 50, 20, 10, 5, 2, 1};
9     int count = 0;
10
11     for (int denom : numbers)
12     {
13         count += n / denom;
14         n %= denom;
15     }
16
17     cout << count << endl;
18
19     return 0;
20 }
```

Accepted	0.003	1.215	View
----------	-------	-------	----------------------

ZOOM TIME:



Висновок: На цій лабораторній роботі я дізнався, що таке рекурсія і як її

застосовувати, навчився використовувати одновимірні і двовимірні масиви. Також дізнався, що таке вказівники і посилання та яка між ними різниця і що краще застосовувати, щоб ефективніше користуватись і керувати пам'яттю та навчився використовувати динамічні масиви.