

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 ШІ-12
Гаврих Юрій Дмитрович

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:

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.

Теоретичні відомості:

[Масиви](#)

[Вказівники](#)

[Посилання](#)

[Структури](#)

Виконання роботи

Task 3 - Lab# programming: VNS Lab 4

Час виконання ~ 30 хв.

- 1) Реалізувати з використанням масиву двонаправлене кільце (перегляд можливий в обидва боки, від останнього елемента можна перейти до першого).
- 2) Роздрукувати отриманий масив, починаючи з K-ого елемента і до K-1 (по кільцю вліво).
- 3) Знищити з кільця перший й останній елементи.
- 4) Роздрукувати отриманий масив, починаючи з K-ого елемента (і до K+1 по кільцю вправо).

Розв'язок:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  void remove_begin_end(int &n,int* arr){
5      for(int i=0;i<n-2;i++){
6          arr[i]=arr[i+1];
7      }
8      n-=2;
9  }
10 void print_arr(int n, int k, int* arr){
11     k--;
12     for(int i=0;i<n;i++){
13         cout<<arr[(i+k)%n]<<" ";
14     }
15     cout<<endl;
16 }
17
18 void print_arr_reverse(int n, int k, int* arr){
19     k--;
20     for(int i=n;i>0;i--){
21         cout<<arr[(i+k)%n]<<" ";
22     }
23     cout<<endl;
24 }
25
26 int main(){
27     int n;
28     cin>>n;
29     int arr[n];
30     for(int i=0;i<n;i++){
31         cin>>arr[i];
32     }
33     int k;
34     cin>>k;
35     print_arr(n,k,arr);
36     remove_begin_end(n,arr);
37     print_arr_reverse(n,k,arr);
38 }
```

Результат виконання:

```

10
1 2 3 4 5 6 7 8 9 10
5
5 6 7 8 9 10 1 2 3 4
6 5 4 3 2 9 8 7

```

Task 4 - Lab# programming: VNS Lab 5

Час виконання ~ 30 хв.

Задано двовимірний масив. Знайти суму елементів першого стовпця без одного останнього елемента, суму елементів другого стовпця без двох останніх, суму елементів третього стовпця без трьох останніх і т.д. Останній стовпець не обробляється. Серед знайдених сум знайти максимальну.

Розв'язок:

```

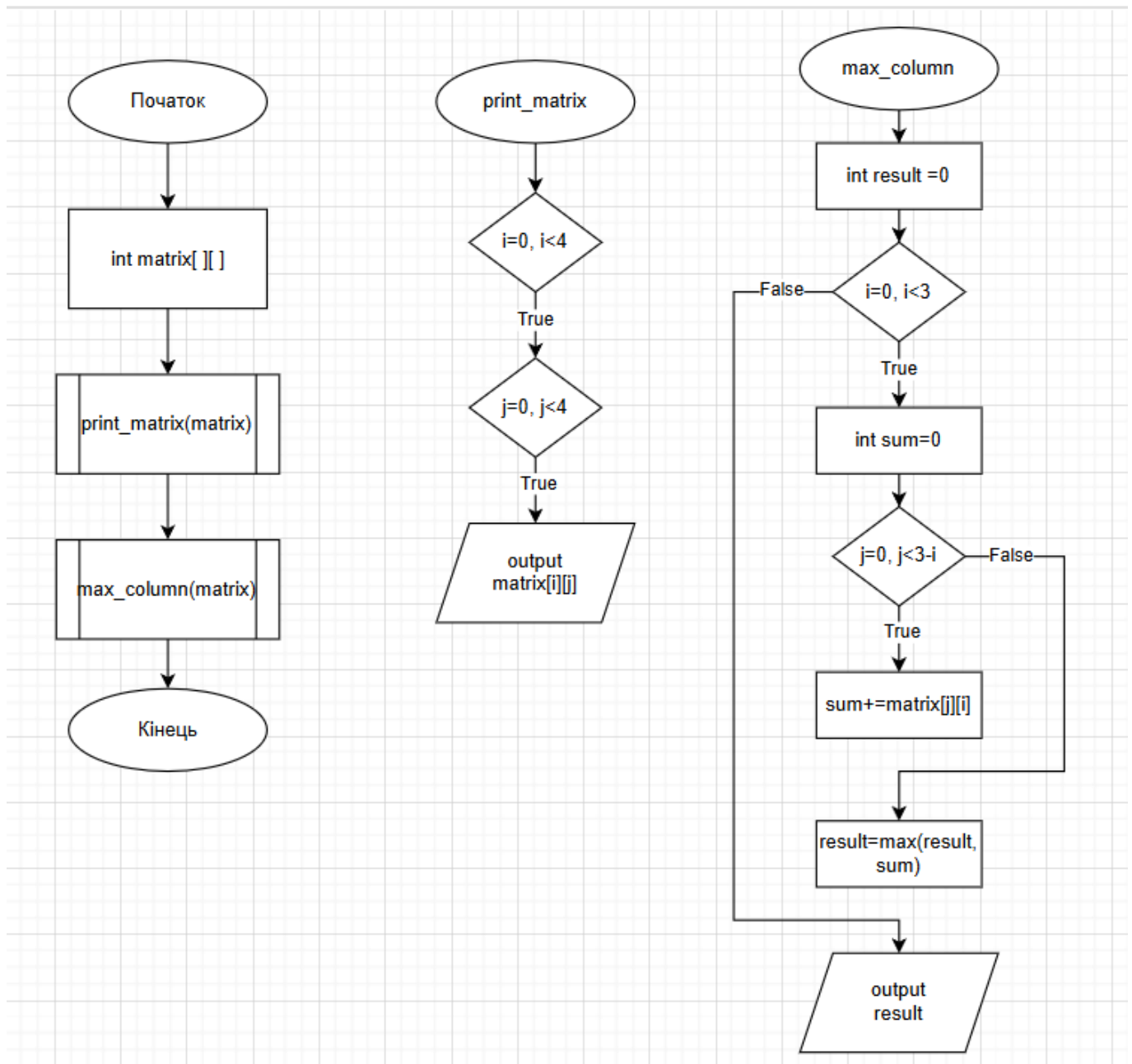
5  #include<bits/stdc++.h>
6  using namespace std;
7
8  void print_matrix(int matrix[4][4]){
9      for(int i=0; i<4; i++){
10         for(int j=0; j<4; j++){
11             cout<<matrix[i][j]<<" ";
12         }
13         cout<<endl;
14     }
15 }
16
17
18 void max_column(int matrix[4][4]){
19     int result=0;
20     for(int i=0;i<3;i++){
21         int sum=0;
22         for(int j=0;j<3-i;j++){
23             sum+=matrix[j][i];
24         }
25         result=max(result,sum);
26     }
27     cout<< result;
28 }
29
30 int main(){
31     int matrix[4][4] = { {1,6,3,9}, {9,0,5,4} , {4,5,1,8}, {1,5,8,2} };
32     print_matrix(matrix);
33     max_column(matrix);
34 }

```

Результат виконання:

1	6	3	9
9	0	5	4
4	5	1	8
1	5	8	2
14			

Блок-схема:



Task 5 - Lab# programming: Algotester Lab 2

Час виконання ~ 20 хв.

Розв'язок:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      int n,x,y,z;
6      cin>>n;
7      vector<int> a(n),b;
8      for(int i=0;i<n;i++){
9          cin>>a[i];
10     }
11     cin>>x>>y>>z;
12     for(int i=0;i<n;i++){
13         if ((a[i]!=x)&&(a[i]!=y)&&(a[i]!=z))
14             b.push_back(a[i]);
15     }
16     int m=b.size();
17     cout<<max(0,m-1)<<endl;
18     for(int i=0;i<m-1;i++){
19         cout<<b[i]+b[i+1]<<" ";
20     }
21 }
```

Результат виконання:

5	6
5 4 3 2 1	1 2 3 4 5 7
4 5 6	4 5 6
2	3
5 3	3 5 10

Task 6 - Lab# programming: Algotester Lab 3

Час виконання ~ 35 хв.

Розв'язок:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  int main(){
5      int n,m;
6      cin>>n;
7      vector<int>a(n),c(n),d;
8      for(int i=0;i<n;i++){
9          cin>>a[i];
10         c[i]=a[i];
11     }
12     cin>>m;
13     vector<int>b(m);
14     for(int i=0;i<m;i++){
15         cin>>b[i];
16         bool flag=true;
17         for(int j:c){
18             if (j==b[i]) flag=false;
19         }
20         if (flag) c.push_back(b[i]);
21     }
22     for(int cur_element:c){
23         bool in_a = false, in_b = false;
24         for(int i:a){
25             if (i==cur_element) in_a=true;
26         }
27         for(int i:b){
28             if (i==cur_element) in_b=true;
29         }
30         if ( (in_a) && (in_b) ) d.push_back(cur_element);
31     }
32     cout<<d.size()<<endl<<c.size();
33
34
35 }
```

Результат виконання:

5	5
1 2 3 4 5	1 2 3 4 5
5	5
4 5 6 7 8	6 7 8 9 10
2	0
8	10

Task 7 - Practice# programming: Class Practice Task

Час виконання ~ 40 хв.

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

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  bool isPalindrome(string str, int left, int right){
5      bool isPal=true;
6      if(left+1<right){
7          isPal=isPalindrome(str,left+1, right-1);
8      }
9      if ( (isPal) && (str[left]==str[right]) ) return true;
10     return false;
11 }
12
13 vector<int> divide(int num){
14     vector<int> a;
15     while(num>0){
16         a.push_back(num%10);
17         num/=10;
18     }
19     return a;
20 }
21 bool isPalindrome(vector<int> &a, int left, int right){
22     bool isPal=true;
23     if(left+1<right){
24         isPal=isPalindrome(a,left+1, right-1);
25     }
26     if ( (isPal) && (a[left]==a[right]) ) return true;
27     return false;
28 }
29
30 int main(){
31     string str;
32     cout<<"String:";
33     cin>>str;
34     int num;
35     cout<<"Number:";
36     cin>>num;
37     vector<int>a = divide(num);
38     cout<<"Is string palindrome? "<< (isPalindrome(str,0,str.size()-1) ? "Yes" : "No" )<<endl;
39     cout<<"Is number palindrome? "<< (isPalindrome(a,0,a.size()-1) ? "Yes" : "No" )<<endl;
40 }
```

Результат виконання:

```
String:abbbccbbba
Number:1234554321
Is string palindrome? Yes
Is number palindrome? Yes
```

```
String:abca
Number:123456
Is string palindrome? No
Is number palindrome? No
```

Task 8 - Practice# programming: Self Practice Task

Час виконання~ 2 год.

[Algotester](#) №249 Бійка у Верховній Раді

Розв'язок:

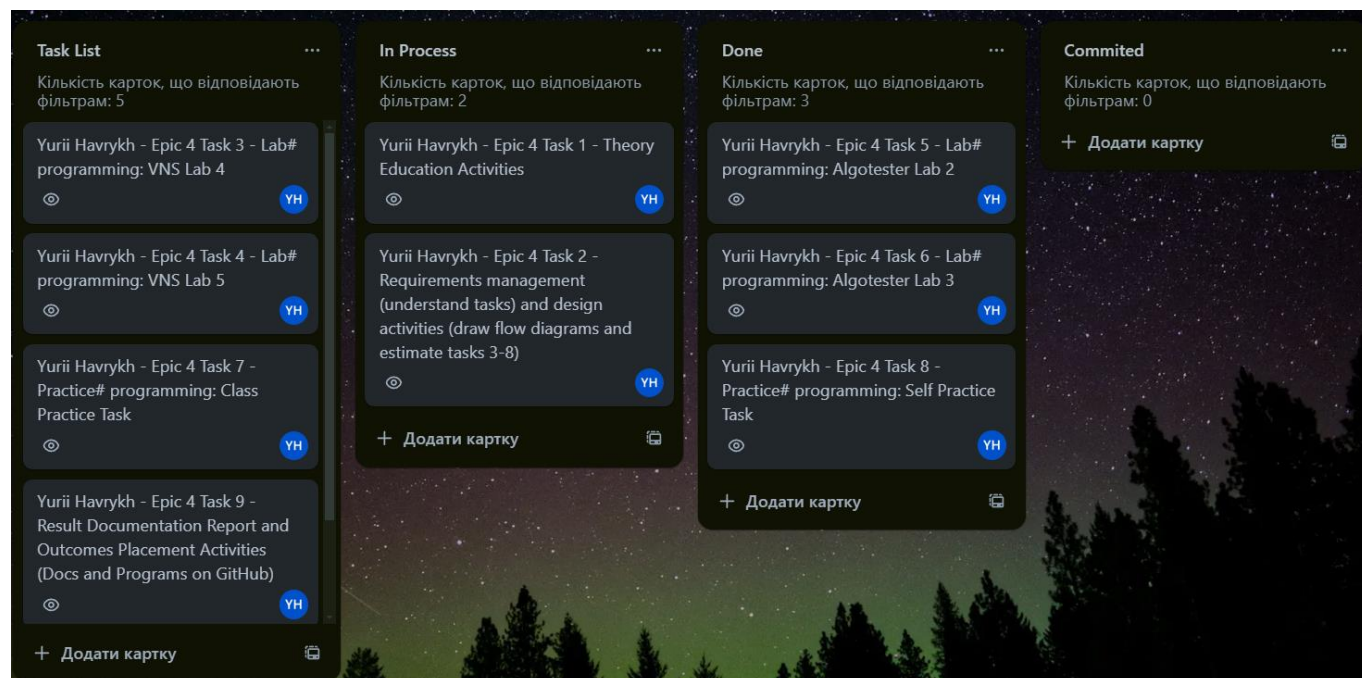

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  void read_agressive(int n,vector<int> &calm){
5      int x;
6      for(int i=0;i<n;i++){
7          cin>>x;
8          calm.push_back(x);
9      }
10 }
11
12 void read_friend(int m,vector<vector<int>> &g){
13     int x,y;
14     for(int i=0;i<m;i++){
15         cin>>x>>y;
16         x--;
17         y--;
18         g[x].push_back(y);
19         g[y].push_back(x);
20     }
21 }
22 void read_fighter(int k, vector<int> &calm, queue<int> &q){
23     int x;
24     for(int i=0;i<k;i++){
25         cin>>x;
26         x--;
27         calm[x]=0;
28         q.push(x);
29     }
30 }
31
32 int bfs(vector<int> &calm, vector<vector<int>> g, queue<int> &q){
33     int rez=0;
34     while(!q.empty()){
35         int cur = q.front();
36         q.pop();
37         for(auto v:g[cur]){
38             if (calm[v]==1){
39                 rez++;
40                 q.push(v);
41             }
42             calm[v]=max(calm[v]-1,0);
43         }
44     }
45     return rez;
46 }
47
48
49 int main() {
50     int n1, n2, m1, m2, k1, k2;
51     cin>>n1>>n2;
52     vector<vector<int>> g1(n1),g2(n2);
53     vector<int> calm1,calm2;
54     read_agressive(n1,calm1);
55     read_agressive(n2,calm2);
56     cin>>m1>>m2;
57     read_friend(m1,g1);
58     read_friend(m2,g2);
59     cin>>k1>>k2;
60     queue<int> q1, q2;
61     read_fighter(k1,calm1,q1);
62     read_fighter(k2,calm2,q2);
63
64
65     int result = bfs(calm1,g1,q1)+bfs(calm2,g2,q2)+k1+k2;
66     cout<<result;
67 }
68
69
70
71
72

```

Task 9 - Result Documentation Report and Outcomes Placement Activities

Час виконання ~ 1,5 год.



Pull Request

Висновок:

В результаті виконання цієї роботи я навчився ефективно використовувати одновимірні та двовимірні масиви. Опанував принципи роботи з посиланнями та вказівниками, а також використав алгоритми обробки та роботи з масивами.