

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 ІІІ-12
Стефанович Євгеній

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.

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

Теоретичні відомості:

У даній 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:

-aCode

- Harvard CS50 lectures+tasks

- chatGPT / Google

Lab# programming: VNS Lab 4

Time estimated: 1h+

Spent: 30min

16.

Реалізувати з використанням масиву чергу (перший прийшов, перший пішов), для чого організувати додавання, знищення елементів у масиві і друк масиву після кожної операції.

```
C: > Users > Eugene > Desktop > epic_4 > G+ vns_lab_4_task_eugenie_stefanovich.cpp > main()
1  #include<iostream>
2  using namespace std;
3
4
5
6  int main() {
7
8      int cher[3];
9      int k = 0;
10
11     for(int i = 0; i<3; i++){
12         cin >> k;
13         cher[i] = k;
14     }
15
16     for (int i = 0; i < 3; i++)
17         cout << cher[i] << " ";
18     cout << endl;
19
20     for(int l=0; l < 3; l++){
21         for (int i = 0; i < 2; i++)
22             cher[i] = cher[i + 1];
23
24         cin >> k;
25         cher[2] = k;
26
27         for (int i = 0; i < 3; i++)
28             cout << cher[i] << " ";
29         cout << endl;
30
31     }
32
33     return 0;
34 }
35
```

```
PS C:\Users\Eugene\Desktop\epic_4\output> cd ..
PS C:\Users\Eugene\Desktop\epic_4\output> & .\
1 2 3
1 2 3
4
2 3 4
5
3 4 5
6
4 5 6
PS C:\Users\Eugene\Desktop\epic_4\output>
```

Lab#
VNS Lab 5

programming:

Time estimated: 2h

Spent: 3h+

16. Задано двовимірний масив $N \times N$. Дозволяється довільно переставляти елементи усередині будь-якого стовпця. Перевірити, чи можна виконавши скінчену кількість перестановок у стовпцях, розташувати на бічній діагоналі елементи так, щоб вони зростали.

```
C: > Users > Eugene > Desktop > epic_4 > vns_lab_5_task_eugenie_stefanovich.cpp > main0
1  #include <iostream>
2  using namespace std;
3
4
5  int main(){
6
7      int N = 3 , wid , inf = 2147483647;
8
9      int mat[3][3];
10
11     for(int i =0; i < N ; i++){
12         for(int l = 0; l < N; l++){
13             cin >> wid;
14             mat[i][l] = wid;
15         }
16     }
17
18     cout << endl;
19
20     for(int i =0; i < N ; i++){
21         for(int l = 0; l < N; l++){
22             cout << mat[i][l] << " ";
23         }
24         cout << endl;
25     }
26     int lmin;
27     int aks = -inf;
28
29     for(int i = N - 1; i > -1; i--){
30         lmin = inf;
31         for(int l = 0; l < N; l++){
32             if(mat[l][i] > aks && mat[l][i] < lmin)
33                 lmin = mat[l][i];
34
35             if(lmin > aks && lmin != inf){
36                 aks = lmin;
37                 cout << aks << endl;
38             }
39             else{
40                 cout << "Не можна";
41                 exit(0);
42             }
43         }
44
45         cout << "Можна";
46
47         return 0;
48     }
49
50 }
```

```
PS C:\Users\Eugene\Desktop\epic_4\output> & .\
1 2 3 4 5 6 7 8 9

1 2 3
4 5 6
7 8 9
3
5
7
Можна
PS C:\Users\Eugene\Desktop\epic_4\output> █
```

Lab# programming: Algotester Lab 2

Time estimated: 15min

Spent: 15 min

У вас є дорога, яка виглядає як N чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

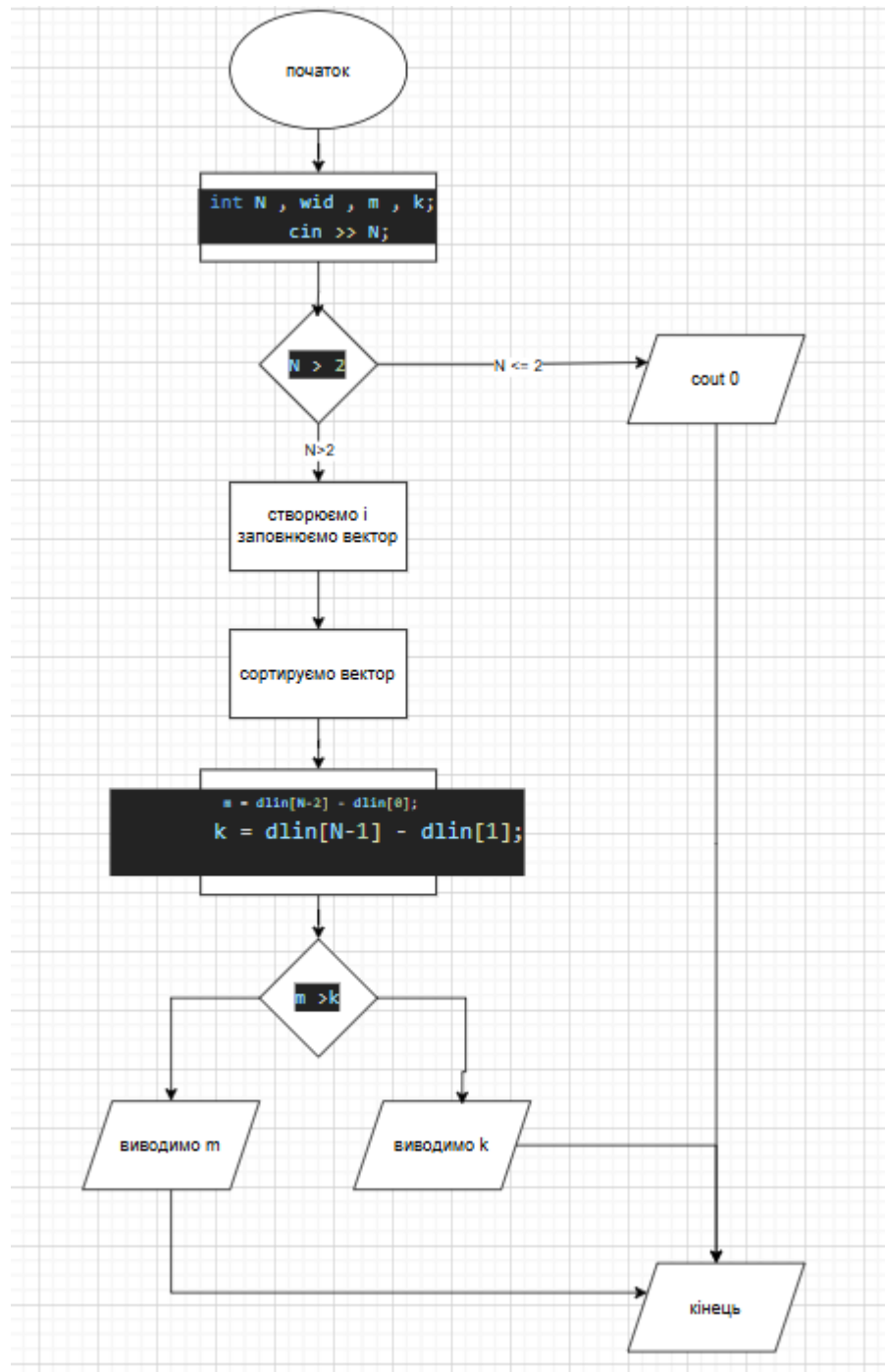
В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

```

C: > Users > Eugene > Desktop > epic_4 > C algotester_lab_2_task_eugenie_stefanovich.cpp >
1  < #include <iostream>
2  < #include <vector>
3  < #include <algorithm>
4
5  using namespace std;
6
7  < int main(){
8      int N , wid , m , k;
9      cin >> N;
10 <     if(N > 2){
11         vector<int> dlin;
12 <         for(int i = 0; i < N; i++){
13             cin >> wid;
14             dlin.push_back(wid);
15         }
16
17         sort(dlin.begin(), dlin.end());
18
19         m = dlin[N-2] - dlin[0];
20         k = dlin[N-1] - dlin[1];
21
22         if(m > k)
23             cout << k;
24         else
25             cout << m;
26     }
27
28 <     else if(N==2){
29         cin >> wid;
30         cin >> wid;
31         cout << 0;
32     }
33 <     else{
34         cin >> wid;
35         cout << 0;
36     }
37
38
39     return 0;
40 }

```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
an hour ago	C++ 23	Accepted	0.003	1.199	View



Lab# programming: Algotester Lab 3

Time estimated: 15min

Spent: 30h

C: > Users > Eugene > Desktop > epic_4 > algotester_lab_3_task_eugenie_stefanovich.cpp > main()					
<pre> 1 #include <iostream> 2 #include <string> 3 4 using namespace std; 5 6 int main() { 7 string S; 8 getline(cin, S); 9 int dlin = S.length(); 10 string result; 11 12 for (int i = 0; i < dlin; i++) { 13 int counter = 1; 14 15 while (i + 1 < dlin && S[i] == S[i + 1]) { 16 counter++; 17 i++; 18 } 19 20 result.push_back(S[i]); 21 22 if (counter > 1) { 23 result += to_string(counter); 24 } 25 } 26 27 cout << result; 28 } 29 </pre>					
Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
39 minutes ago	C++ 23	Accepted	0.003	1.211	View

Practice# programming: Class Practice Task

Time estimated: 45min

Spent: 25min

Задача

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»).


```

G: > Users > Eugene > Desktop > epic_4 > practice_work_team_tasks_eugenie_stefanovich.cpp > isPalindrome(int)
1  #include <iostream>
2
3  using namespace std;
4
5  bool isPalindrome(const string& s, int start, int end){
6      if(start >= end)
7          return true;
8      if(s[start] != s[end])
9          return false;
10     return isPalindrome(s, start+1, end-1);
11 }
12
13 bool isPalindrome(int n) {
14     if (n < 0)
15         return false;
16
17     int original = n;
18     int reversed = 0;
19
20     while (n > 0) {
21         int p = n % 10;
22         reversed = reversed * 10 + p;
23         n /= 10;
24     }
25
26     return original == reversed;
27 }
28
29 int main(){
30     string s;
31     int n;
32     cout << "Строка: ";
33     getline(cin, s);
34     cout << endl;
35     cout << "Число: ";
36     cin >> n;
37     cout << endl;
38
39     cout << "Строка є паліндромом? - " << (isPalindrome(s, 0, s.length()-1) ? "Так" : "Hi") << endl;
40     cout << "Число є паліндромом? - " << (isPalindrome(n) ? "Так" : "Hi") << endl;
41 }

```

```

Число є паліндромом? - Так
PS C:\Users\Eugene\Desktop\epic_4\output> cd 'c:\Users\E
PS C:\Users\Eugene\Desktop\epic_4\output> & .\practice_
Строка: ABOBA
Число: 1234321

Строка є паліндромом? - Так
Число є паліндромом? - Так
PS C:\Users\Eugene\Desktop\epic_4\output>

```

Practice# programming: Self Practice Task

Time estimated: 20min

Spent: 45min

```
C: > Users > Eugene > Desktop > epic_4 > C
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      int n,m;
8      cin >> m;
9      cin >> n;
10     if((m*n)%2 == 0)
11         cout << "Dragon";
12     else
13         cout << " Imp";
14
15
16     return 0;
17 }
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.002	1.047	Перегляд

Висновок: В процесі виконання лабораторної роботи я навчився використовувати одновимірні та двовимірні масиви для зберігання і впорядкування даних, що покращує доступ до великого обсягу інформації. Також я ознайомився з поняттями вказівників і посилань, що дозволяє ефективно управляти пам'яттю і використовувати динамічні масиви. Окрім того, я досліджував структури даних та алгоритми обробки масивів, що сприяє написанню оптимізованих і масштабованих програм.