

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



Звіт

про виконання лабораторних та практичних робіт блоку № 3

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

Алготестер Лабораторної Роботи № 2,3

Практичних Робіт до блоку № 4

Виконав:

Студент групи ШІ-12

Токарік Сергій

Львів 2024

Тема роботи

1. Використання одновимірних та двовимірних масивів, а також динамічних масивів.
2. Використання інформаційних структур.
3. Використання вказівників, посилань.

Мета роботи

1. Навчитись створювати одновимірні, двовимірні та динамічні масиви користуватись ними.
2. Навчитись користуватись структурами.
3. Навчитись користуватись вказівниками та посиланнями.

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>

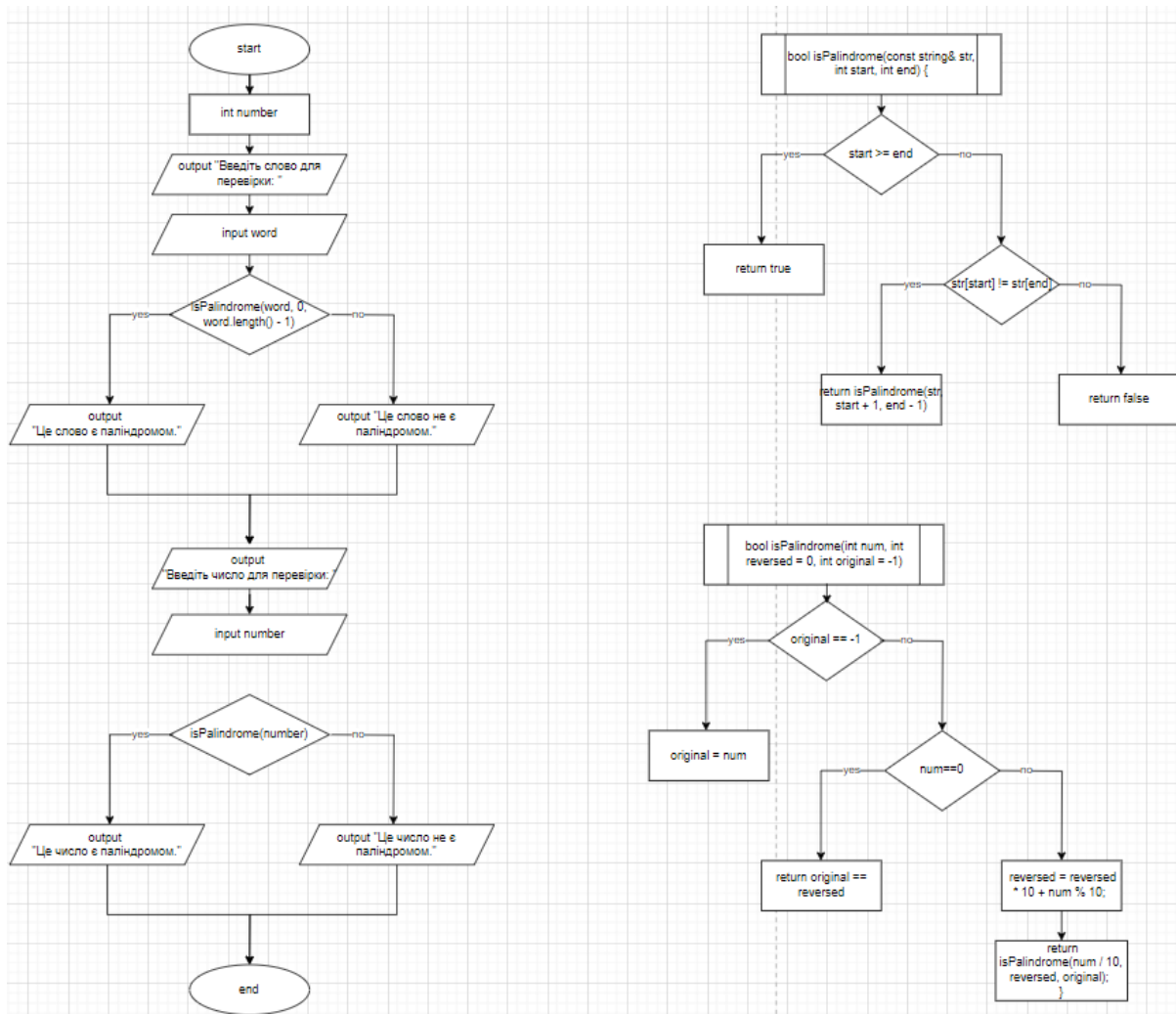
4. Посилання

<https://acode.com.ua/urok-92-posylannya/#toc-0>

Виконання робіт

Task 2 - Requirements management (understand tasks) and design activities
(draw flow diagrams and estimate task) (1 год 40хв)

VNS Lab 5



Task 3 - Lab# programming: VNS Lab 4(10хв)

24.

- 1) Реалізувати з використанням масиву однонаправлене кільце (перегляд можливий зліва направо, від останнього елемента можна перейти до першого).
- 2) Роздрукувати отриманий масив, починаючи з K-ого елемента й до K-1.
- 3) Впорядкувати елементи за зростанням
- 4) Знищити з кільця парні елементи.
- 5) Роздрукувати отриманий масив, починаючи з K-ого елемента і до K-1.

```
1  #include <iostream>
2  #include <algorithm>
3
4  using namespace std;
5
6  void print(const int massive[], int realSize, int startIndex) {
7      for (int i = 0; i < realSize; i++) {
8          cout << massive[(startIndex + i) % realSize] << " ";
9      }
10     cout << endl;
11 }
12
13 int main() {
14     const int N = 100;
15     int massive[N];
16
17     int realSize;
18     cout << "Введіть кількість елементів масиву (не більше 100): ";
19     cin >> realSize;
20     if (realSize > N) {
21         cout << "Помилка: розмір масиву перевищує максимальний розмір (" << N << ")." << endl;
22         return 1;
23     }
24
25     cout << "Введіть елементи масиву: ";
26     for (int i = 0; i < realSize; i++) {
27         cin >> massive[i];
28     }
29
30     int K;
31     cout << "Введіть елемент K: ";
32     cin >> K;
33
34     cout << "Масив зі зсувом K: ";
35     print(massive, realSize, K);
36
37     sort(massive, massive + realSize);
38     cout << "Відсортований масив: ";
39     for (int i = 0; i < realSize; i++) {
40         cout << massive[i] << " ";
41     }
42     cout << endl;
43
44     int newSize = 0;
45     for (int i = 0; i < realSize; i++) {
46         if (massive[i] % 2 != 0) {
47             massive[newSize++] = massive[i];
48         }
49     }
50     realSize = newSize;
51
52     cout << "Масив після видалення парних чисел: ";
53     print(massive, realSize, K);
54
55     return 0;
56 }
57
58
```

24. Визначити скільки елементів двовимірного масиву більші від будь-якого елемента на головній діагоналі.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 #define N 10
6
7 using namespace std;
8
9 int count_biggers(int** a) {
10     int max_on_diag = a[0][0];
11     int count = 0;
12
13     for (size_t i = 1; i < N; i++) {
14         if (a[i][i] > max_on_diag) {
15             max_on_diag = a[i][i];
16         }
17     }
18
19     for (size_t i = 0; i < N; i++) {
20         for (size_t j = 0; j < N; j++) {
21             if (a[i][j] > max_on_diag) {
22                 count++;
23             }
24         }
25     }
26     return count;
27 }
28
29 int main() {
30     srand(static_cast<unsigned>(time(0)));
31
32     int** arr = new int*[N];
33     for (size_t i = 0; i < N; i++) {
34         arr[i] = new int[N];
35     }
36
37     for (size_t i = 0; i < N; i++) {
38         for (size_t j = 0; j < N; j++) {
39             arr[i][j] = rand() % 100;
40         }
41     }
42
43     for (size_t i = 0; i < N; i++) {
44         for (size_t j = 0; j < N; j++) {
45             if (i == j) {
46                 cout << "" << arr[i][j] << "\t";
47             } else {
48                 cout << arr[i][j] << "\t";
49             }
50         }
51         cout << endl;
52     }
53
54     cout << "Кількість елементів, більших за максимальний елемент головної діагоналі: "
55          << count_biggers(arr) << endl;
56
57     for (size_t i = 0; i < N; i++) {
58         delete[] arr[i];
59     }
60     delete[] arr;
61
62     return 0;
63 }
```

Task 5 - Lab# programming: Algotester Lab 2

Lab 2v2

Обмеження: 1 сек., 256 МіБ

У вас є масив r розміром N . Також вам дано 3 цілих числа.

Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей

масив у масив сум, розміром $N_{\text{new}} - 1$ (розмір нового масиву після видалення елементів), який

буде відображати суми сусідніх елементів нового масиву.

Далі необхідно вивести масив сум на екран.

```
#include <iostream>

using namespace std;

int main() {
    int N;
    cin >> N;

    int r[N];
    for (int i = 0; i < N; i++) {
        cin >> r[i];
    }

    int a, b, c;
    cin >> a >> b >> c;

    int M = 0;
    int r1[N];
    for (int i = 0; i < N; i++) {
        if (r[i] != a && r[i] != b && r[i] != c) {
            r1[M] = r[i];
            M++;
        }
    }

    int sums[M - 1];
    for (int i = 0; i < M - 1; i++) {
        sums[i] = r1[i] + r1[i + 1];
    }

    cout << M - 1 << "\n";
    for (int i = 0; i < M - 1; i++) {
        cout << sums[i] << " ";
    }

    cout << endl;

    return 0;
}
```

Task 6 - Lab# programming: Algotester Lab 3

Lab 3v2

Обмеження: 1 сек., 256 МіБ

Вам дано 2 масиви розміром N та M . Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одно-
часно, у другому кількість унікальних елементів в обох масивах разом.

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  string compressString(const string& s) {
7      string result;
8      int count = 1;
9
10     for (size_t i = 1; i <= s.size(); ++i) {
11
12         if (i == s.size() || s[i] != s[i - 1]) {
13             result += s[i - 1];
14             if (count > 1) {
15                 result += to_string(count);
16             }
17             count = 1;
18         } else {
19             ++count;
20         }
21     }
22
23     return result;
24 }
25
26 int main() {
27     string s;
28     cin >> s;
29
30     string compressed = compressString(s);
31     cout << compressed << endl;
32
33     return 0;
34 }
35
```

Перевірка чи слово або число є паліндромом

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»).

Мета Задачі

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

Вимоги:

1. Визначення функції:
 - a. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
 - a. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
 - a. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - b. *bool isPalindrome(ціле число);*
4. Рекурсія:
 - a. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

Кроки реалізації

- Визначте та реалізуйте рекурсивну функцію *isPalindrome* для рядків.
- Визначте та реалізуйте перевантажену функцію *isPalindrome* для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.


```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5
6  bool isPalindrome(const string& str, int start, int end) {
7      if (start >= end) {
8          return true;
9      }
10     if (str[start] != str[end]){
11         return false;
12     }
13     return isPalindrome(str, start + 1, end - 1);
14 }
15
16
17 bool isPalindrome(int num, int reversed = 0, int original = -1) {
18     if (original == -1) original = num;
19     if (num == 0) return original == reversed;
20
21     return isPalindrome(num / 10, reversed, original);
22 }
23
24
25
26 int main() {
27     string word;
28     int number;
29
30     cout << "Введіть слово для перевірки: ";
31     cin >> word;
32     if (isPalindrome(word, 0, word.length() - 1)) {
33         cout << "Це слово є паліндромом." << endl;
34     }
35
36     else {
37         cout << "Це слово не є паліндромом." << endl;
38     }
39
40     cout << "Введіть число для перевірки: ";
41     cin >> number;
42     if (isPalindrome(number)) {
43         cout << "Це число є паліндромом." << endl;
44     }
45
46     else {
47         cout << "Це число не є паліндромом." << endl;
48     }
49
50     return 0;
51 }
52

```

Task 8 - Practice# programming: Self Practice Task

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int n;
7      cout << "Введіть кількість чисел: ";
8      cin >> n;
9
10     double numbers[n];
11
12     cout << "Введіть " << n << " чисел:" << endl;
13     for (int i = 0; i < n; ++i) {
14         cout << "Число " << (i + 1) << ": ";
15         cin >> numbers[i];
16     }
17
18     double sum = 0;
19     for (int i = 0; i < n; ++i) {
20         sum += numbers[i];
21     }
22     double average = sum / n;
23
24     cout << "Середнє значення: " << average << endl;
25
26     return 0;
27 }
28
```

Зустрічі з командою





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

Висновок

В ході даного епіку я навчився1 використовувати одновимірні, двовимірні та динамічні масиви, структури даних а також вказівники, краще навчився використовувати функції та в загальному покращив свої знання в сфері програмування .