

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: «Мови та парадигми програмування»

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт №4

Виконала:

Студентка групи ІІІ-11

Сподарик Софія Романівна

Львів 2023

Тема роботи:

Прості структури даних. Одновимірні масиви. Двовимірні масиви.
Алгоритми обробки

Мета роботи:

Ознайомитися з структурами даних та поглибити знання у працюванні з масивами. Практично застосувати знання, пишучи коди до практичних та лабораторних роботи.

Теоретичні відомості:

Теоретичні відомості з переліком важливих тем та індивідуальних план опрацювання їх:

- Тема №1: Одновимірні масиви

- Джерела Інформації
 - [Урок №77. Масиви](#)
 - [Урок №78. Фіксовані масиви](#)
 - [Урок №79. Масиви і цикли](#)
 - [C++ Arrays](#)
 - <https://www.youtube.com/watch?v=ULdbOaMBPYc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=41>
 - Матеріал з ВНС №4
- Що опрацьовано:
 - одновимірні масиви та їхнє застосування
- Статус: Ознайомлена
- Початок опрацювання теми: 28.11.2023
- Звершення опрацювання теми: 06.12.2023

- Тема №2: Двовимірні масиви

- Джерела Інформації
 - [Урок №81. Двовимірні масиви](#)
 - [Урок №90. Динамічні масиви](#)
 - [Multidimensional Arrays](#)
 - [C++ Function and Array](#)

- <https://www.youtube.com/watch?v=V2g3B9Zbh4Q&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=42>
- <https://www.youtube.com/watch?v=1DtZCv7xfb8&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=43>
- Матеріал у ВНС №5
- Що опрацьовано:
 - багатовимірні масиви та їхнє застосування
- Статус: Ознайомлена
- Початок опрацювання теми: 28.11.2023
- Звершення опрацювання теми: 06.12.2023

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1: VNS Lab 4

- Варіант завдання: 20
- Деталі завдання:
 - 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 К-ого елемента і до К-1 (по кільцю вліво).
 - 3) Додати в кільце після елементів з індексами кратними 5 елементи, які дорівнюють 0.
 - 4) Роздрукувати отриманий масив, починаючи з К-ого елемента (і до К+1 по кільцю вправо).
- Важливі деталі для врахування в імплементації програми :
 - 1) При виконанні роботи використовуються статичні масиви. Для організації статичних масивів із псевдо змінними межами необхідно оголосити масив досить великої довжини, наприклад, 100 елементів:

```
int N=100;
```

```
int a[N];
```

Потім користувач вводить реальну довжину масиву (не більше N) і працює з масивом тієї довжини, що він сам вказав. Інші елементи (хоча пам'ять під них і буде виділена) не розглядаються.

2) При зменшенні або збільшенні довжини масиву необхідно змінювати його реальну довжину.

Завдання №2: VNS Lab 5

- Варіант завдання: 20
- Деталі завдання: Знайти мінімальний з неповторюваних елементів двовимірного масиву.

Завдання №3: Algotester Lab 2

- Варіант завдання: 1
- Деталі завдання:

Lab 2v1

Обмеження: 1 сек., 256 МБ

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?

Вхідні дані

У першому рядку ціле число N - кількість чисел
У другому рядку масив r , який складається з N цілих чисел

Вихідні дані

Єдине ціле число m - мінімальна втома, яку можна отримати

Обмеження

$1 \leq N \leq 10^5$
 $0 \leq r_i \leq 10^5$

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
5 1 2 2 4 4	2
5 1 2 2 2 4	1
5 1 3 3 4 4	1
1 5	0

Деталі до завдання №3

Завдання №4: Algotester Lab 3

- Варіант завдання: 3
- Деталі завдання:

Lab 3v3

Обмеження: 1 сек., 256 MiB

Вам дана стрічка s .

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

Вхідні дані

У першому рядку стрічка S

Вихідні дані

Стрічка $S_{compressed}$

Обмеження

$$1 \leq |S| \leq 10^5$$

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
AAABBBVCQQQ	A4B3CQ4
AA	A2
A	A

Примітки

Великі та маленькі букви вважаються різними.

Деталі до завдання №4

Завдання №5: Class Practice

- Деталі завдання та важливі деталі для врахування в імplementації програми:

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:

- Визначення функції:
 - Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
- Приклад визначення функції:
 - bool isPalindrome(const string& str, int start, int end);*
- Перевантаження функції:
 - Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
 - bool isPalindrome(ціле число);*
- Рекурсія:
 - 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. Перетворити ціле число на рядок і використовуйте рядкову версію функції, щоб перевірити, чи це паліндром.

Деталі до завдання №5

Завдання №6: Self Practice

- Деталі завдання:

Юний художник

Обмеження: 1 сек., 256 МБ

Юний художник Зеник має n послідовно розташованих білих комірок, пронумерованих від 1 до n зліва направо, які можна фарбувати.
Він виконує m фарбувань такого вигляду: (l, r, c) — зафарбувати усі комірки від l до r включно в колір c . Кожна комірка пофарбована тільки в останній колір, яким фарбувалася відповідна комірка.
Вам потрібно відновити кінцеву картину, тобто визначити кольори усіх комірок після виконання всіх запитів.

Вхідні дані

У першому рядку задано два цілих числа n і m — кількість комірок та кількість фарбувань.
У наступних m рядках задано по 3 цілих числа l, r і c — відповідне фарбування.

Вихідні дані

У єдиному рядку виведіть n цілих чисел через пробіл — кольори комірок після усіх фарбувань.
Вважайте, що білий колір — це 0.

Обмеження

$1 \leq n, m, c \leq 10^5$,
 $1 \leq l \leq r \leq n$.

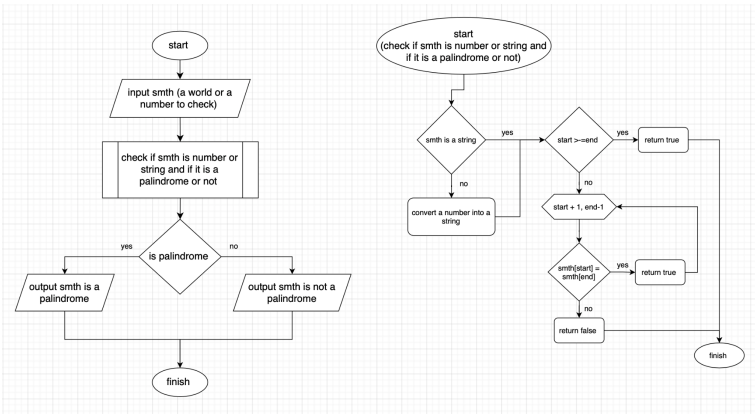
Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
7 4 1 4 7 6 6 1 4 6 2 5 7 1	7 7 7 2 1 1 1

Деталі до завдання №6

2. Дизайн виконання завдань:
Програма № 5

- Блок-схема:



Блок-схема до програми №5

3. Код програм: Завдання №1: VNS Lab 4

```
1 #include <iostream>
2
3 using namespace std;
4
5 struct RingArray {
6     static const int N = 100;
7     int elements[N];
8     int realSize;
9 };
10
11 void printRing(const RingArray& ring, int startIdx, int endIdx) {
12     for (int i = startIdx; i != endIdx; i = (i + 1) % ring.realSize) {
13         cout << ring.elements[i] << " ";
14     }
15     cout << endl;
16 }
17
18 void addZerosAfterMultiplesOfFive(RingArray& ring) {
19     for (int i = 0; i < ring.realSize; ++i) {
20         if (ring.elements[i] % 5 == 0) {
21             for (int j = ring.realSize - 1; j > i; --j) {
22                 ring.elements[(j + 1) % RingArray::N] = ring.elements[j];
23             }
24             ring.elements[(i + 1) % RingArray::N] = 0;
25             ++ring.realSize;
26             ++i;
27         }
28     }
29 }
30
```

```
30
31 int main() {
32     RingArray ring;
33
34     cout << "Enter the real size of the array (not more than " << RingArray::N << "): ";
35     cin >> ring.realSize;
36
37     cout << "Enter array elements: ";
38     for (int i = 0; i < ring.realSize; ++i) {
39         cin >> ring.elements[i];
40     }
41
42     cout << "Original array: ";
43     printRing(ring, 0, ring.realSize);
44
45     int startIdx;
46     cout << "Enter the starting index for printing the array: ";
47     cin >> startIdx;
48
49     cout << "Array printed to the left from index " << startIdx << " ": ";
50     printRing(ring, startIdx, (startIdx - 1 + ring.realSize) % ring.realSize);
51
52     addZerosAfterMultiplesOfFive(ring);
53
54     cout << "Array after adding zeros: ";
55     printRing(ring, 0, ring.realSize);
56
57     int startIdxAfterChanges;
58     cout << "Enter the starting index for printing the array after changes: ";
59     cin >> startIdxAfterChanges;
60
61     cout << "Array printed to the right from index " << startIdxAfterChanges << " ": ";
62     printRing(ring, startIdxAfterChanges, (startIdxAfterChanges + 1) % ring.realSize);
63
64     return 0;
65 }
```

Код до завдання №1

Завдання №2: VNS Lab 5

```
1 #include <iostream>
2 #include <set>
3
4 using namespace std;
5
6 int main() {
7     int rows, columns;
8     cout << "Введіть кількість рядків: ";
9     cin >> rows;
10    cout << "Введіть кількість стовпців: ";
11    cin >> columns;
12
13    int** array = new int*[rows];
14    for (int i = 0; i < rows; ++i) {
15        array[i] = new int[columns];
16    }
17
18    cout << "Введіть елементи двовимірного масиву:" << endl;
19    for (int i = 0; i < rows; ++i) {
20        for (int j = 0; j < columns; ++j) {
21            cout << "Елемент [" << i << "][" << j << "]: ";
22            cin >> array[i][j];
23        }
24    }
25
26    set<int> uniqueElements;
27
28    for (int i = 0; i < rows; ++i) {
29        for (int j = 0; j < columns; ++j) {
30            uniqueElements.insert(array[i][j]);
31        }
32    }
33
34    int minElement = *uniqueElements.begin();
35
36    cout << "Мінімальний з неповторюваних елементів: " << minElement << endl;
37
38    for (int i = 0; i < rows; ++i) {
39        delete[] array[i];
40    }
41    delete[] array;
42
43    return 0;
44 }
```

Код до завдання №2

Завдання №3: Algotester Lab 2

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 int main() {
8     int n;
9     cin >> n;
10
11     vector<int> array(n);
12     for (int i = 0; i < n; ++i) {
13         cin >> array[i];
14     }
15
16     if (n == 1)
17         cout << "0";
18     else {
19         sort(array.begin(), array.end());
20         if (n == 2)
21             cout << array[0];
22         else if (array[1] - array[0] > array[n - 1] - array[n - 2])
23             cout << array[n - 1] - array[1];
24         else
25             cout << array[n - 2] - array[0];
26     }
27     return 0;
28 }
```

Код до завдання №3

Завдання №4: Algotester Lab 3

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main()
7 {
8     string s;
9     cin >> s;
10
11     int count = 1;
12     char letter = s[0];
13
14     for (int i = 1; i <= s.length(); i++)
15     {
16         if (s[i] == letter)
17             count++;
18         else
19         {
20             cout << letter;
21             if (count > 1)
22                 cout << count;
23
24             count = 1;
25             letter = s[i];
26         }
27     }
28
29     return 0;
30 }
31
```

Код до завдання №4

Завдання №5: Class Practice

```
1 #include <iostream>
2 #include <string>
3 #include <algorithm>
4
5 using namespace std;
6
7
8 bool palindrome(const string& smth, int start, int end) {
9     if (start >= end) {
10         return true;
11     }
12
13     if (smth[start] == smth[end]) {
14         return palindrome(smth, start + 1, end - 1);
15     }
16     return false;
17 }
18
19
20 bool palindrome(int number) {
21     string strNumber = to_string(number);
22     return palindrome(strNumber, 0, strNumber.length() - 1);
23 }
24
25
26 int main() {
27     string smth;
28     cout << "Enter smth: ";
29     cin >> smth;
30
31     if (palindrome(smth, 0, smth.length() - 1)) {
32         cout << smth << " is a palindrome." << endl;
33     } else {
34         cout << smth << " is not a palindrome." << endl;
35     }
36
37     return 0;
38 }
```

Код до завдання №5

Завдання №6: Self Practice

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int n, m;
8      cin >> n >> m;
9
10     vector<int> colors(n, 0);
11
12     for (int i = 0; i < m; ++i) {
13         int l, r, c;
14         cin >> l >> r >> c;
15
16         for (int j = l - 1; j < r; ++j) {
17             colors[j] = c;
18         }
19     }
20
21     for (int i = 0; i < n; ++i) {
22         cout << colors[i] << " ";
23     }
24
25     return 0;
26 }
```

Код до завдання №6

4. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №2: VNS Lab 5

```
0_2_task_1_variant_20_sofia_spodatyk
Введіть кількість рядків: 2
Введіть кількість стовпців: 2
Введіть елементи двовимірного масиву:
Елемент [0][0]: 1
Елемент [0][1]: 3
Елемент [1][0]: 4
Елемент [1][1]: 5
Мінімальний з неповторюваних елементів: 1
mac@MACs-MBP epic 4 %
```

Результати коду до завдання №2

Час затрачений на виконання завдання: 1 год

Завдання №3: Algotester Lab 2

```
mac@MACs-MBP epic3 % cd "/Users/mac/Desktop/projects/lpnu/epic
ia_spodatyk.cpp -o algotester_lab_2_task_1_sofia_spodatyk &&
/"algotester_lab_2_task_1_sofia_spodatyk
5
1 2 2 4 4
2
```

Результати коду до завдання №3

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
36 minutes ago	C++	Accepted	0.003	2.043	View

Прийнято алготестером завдання

Час затрачений на виконання завдання: 1 год

Завдання №4: Algotester Lab 3

```
mac@MACs-MBP epic3 % cd "/Users/mac/Desktop/projects/lpnu/epic3/" && g++ a
<& g++ algotester_lab_2_task_1_sofiia_podaryk.cpp -o algotester_lab_2_tas
b_2_task_1_sofiia_spodaryk && "/Users/mac/Desktop/projects/lpnu/epic3/"alg
otester_lab_2_task_1_sofiia_spodaryk
AAABBBBGGAAAC
A3B5GA3C%
mac@MACs-MBP epic3 %
```

Результати коду до завдання №4

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
32 minutes ago	C++	Accepted	0.003	2.164	View

Прийнято алготестером завдання

Час затрачений на виконання завдання: 1.5 год

Завдання №5: Class Practice

```
mac@MACs-MBP epic 4 % cd "/Users/
aryk.cpp -o practice_work_task_1
task_1_sofiia_spodaryk
Enter smth: level
level is a palindrome.
mac@MACs-MBP epic 4 % cd "/Users/
aryk.cpp -o practice_work_task_1
task_1_sofiia_spodaryk
Enter smth: sun
sun is not a palindrome.
```

Результат коду до завдання №5

Час затрачений на виконання завдання: 1 год

Завдання №6: Self Practice

```
7 4
1 4 7
6 6 1
4 6 2
5 7 1
7 7 7 2 1 1 1 %
mac@MACs-MBP epic 4 %

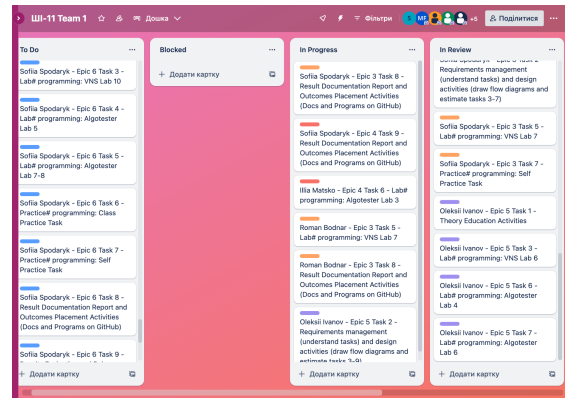
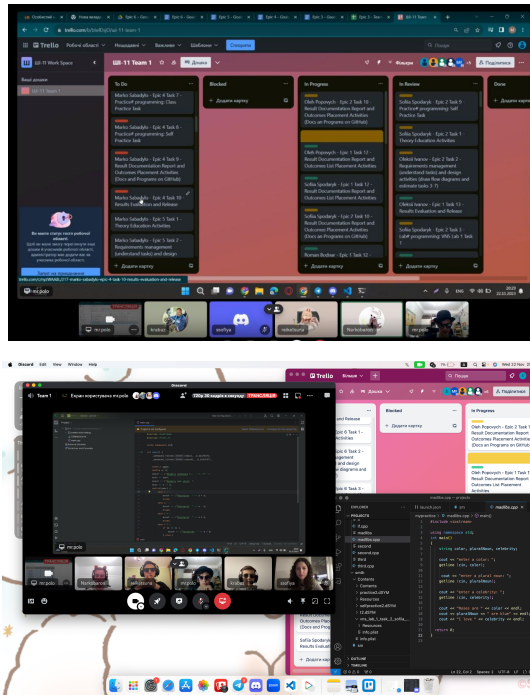
8 6
8 3 4
7 6 3
1 2 8
5 6 7
3 5 4
2 6 1
8 1 1 1 1 1 0 0 %
mac@MACs-MBP epic 4 %
```

Результати коду до завдання №6

Час затрачений на виконання завдання: 20 хв

5. Кооперація з командою:

- Мали зустрічі оффлайн 22.11 та 29.11, де обговорювали прогрес по задачах, допомагали одне одному і розібрали складні моменти.
- Скрін з зустрічі по обговоренню задач Епіку та Скрін прогресу по Тrello



Скріни зі зустрічей з командою

- Скрін з 2 коментарями від учасників команди на пул реквестів з Рев'ю Роботи

Висновок:

На цій лабораторній роботі я дізналась про найпростіші структури даних, такі як одновимірні і двовимірні масиви, а також вектори. Крім цього я зрозуміла їхній принцип і основні алгоритми щодо їх використання. Щоб закріпити вивчений матеріал нам було дано декілька програм з ВНС і Algotester.