



## 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

**Виконала:**

Студентка групи ШІ-13

Щербан Ярина Олегівна

## **Тема:**

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

## **Мета:**

Навчитися створювати масиви, працювати з ними та задавати їх у різних вимірах. Дізнатись про вказівники та посилання. Розібрати класи пам'яті у мові C++. Навчитись використовувати різні алгоритми сортування у відповідних ситуаціях. Створення структур.

## **Теоретичні відомості:**

1. Класи пам'яті у C++
  - Статична пам'ять.
  - Динамічна пам'ять.
  - Поняття стеку.
  - Виділення та вивільнення пам'яті.
3. Одновимірні Масиви
4. Вказівники та Посилання
5. Двовимірні Масиви
6. Динамічні Масиви
7. Структури Даних:
8. Вкладені Структури:
9. Використання структур
10. Алгоритми обробки та робота з масивами та структурами:

## **Опрацювання теоретичного матеріалу :**

1. Вивчення мови C++ за допомогою сайтів : <https://www.w3schools.com/>, <https://acode.com.ua/>
2. Робота з блок-схемами та Draw іо <https://www.programiz.com/article/flowchart-programming>
3. Опрацювала відео щодо роботи з масивами <https://youtu.be/ULdbOaMBPYc?feature=shared>

## **Виконання роботи:**

1) *Опрацювання завдання та вимог до програми та середовища*

### **Завдання №1 Епiк 4 : Практичне завдання :**

Опис задачі : 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»).

### **Умови задачі :**

1. Визначення функції: Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції: *bool isPalindrome(const string& str, int start, int end);*

3. Перевантаження функцій: Перевантажте функцію *isPalindrome* для роботи з цілими значеннями. *bool isPalindrome*(ціле число);
4. 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.

### **Завдання №2 VNS Lab 4 - Task 1 – Variant 5 :**

#### Опис задачі :

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити елементи кратні 7.
- 4) Додати після кожного непарного елемента масиву елемент зі значенням 0.
- 5) Роздрукувати отриманий масив.

### **Завдання №4 VNS Lab 5 - Task 1 – Variant 21 :**

Опис задачі : Використовуючи функції, розв'язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр.

Умови задачі : Знайти максимальний з повторюваних елементів двовимірного масиву.

### **Завдання №4 Algotester Lab 2 - Variant 1 :**

Опис задачі : 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 елемента. 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?

Умови задачі : Ввести у першому рядку ціле число N - кількість чисел. У другому рядку масив r, який складається з N цілих чисел.

### **Завдання №5 Algotester Lab 3 - Variant 2 :**

Опис задачі : 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 одночасно, у другому кількість унікальних елементів в обох масивах разом.

Умови задачі : Вивести у першому рядку одне ціле число - кількість елементів, які наявні в обох масивах одночасно. У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об'єднанням двох даних).

### **Завдання №6 Algotester Lab 3 - Variant 3:**

Опис задачі : 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.

## Завдання №7 Algotester : День програміста

Опис задачі : Нарешті ми його дочекалися, 256-го дня в році (дня програміста)... Кожен зі студентів факультету прикладної математики та інформатики святкували його по-різному: дехто вдома, дехто в «Picasso», дехто в гуртожитку... Не дивно, що Зеник із Марічкою залишилися святкувати його в гуртожитку. Вони організували «mega party» :-). Свято вдалося...

Наступного дня, гуляючи з Марічкою, Зенику стало цікаво, скільки ж було випито різної випивки?!

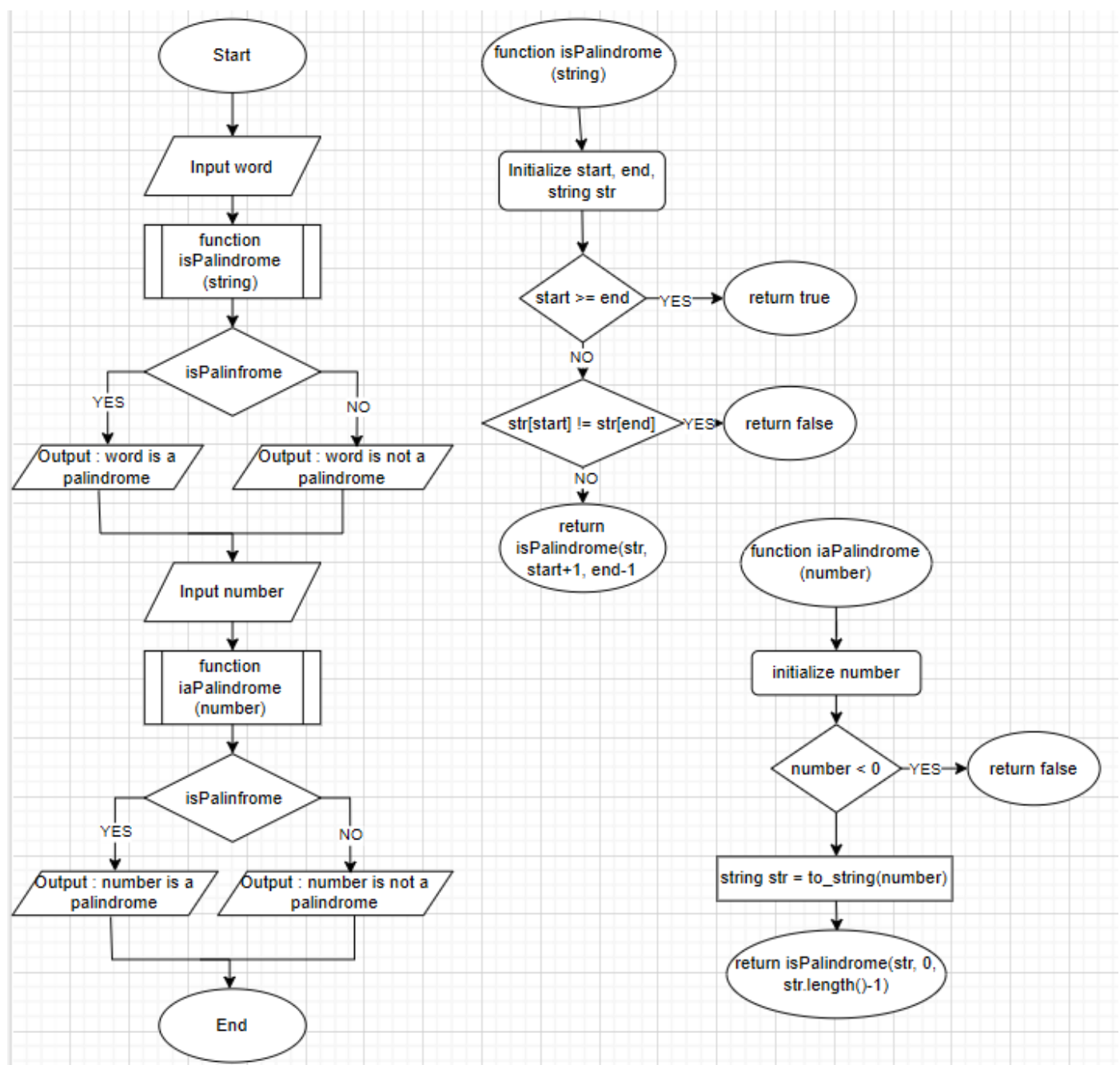
Марічка змогла згадати nn назв випивок, а Зеник — mm.

Вам потрібно написати програму, яка порахує скільки ж було різної випивки на святі, яку змогли згадати Зеник з Марічкою.

2) Дизайн та планована оцінка часу виконання завдань

## Завдання №1 Епік 4 : Практичне завдання :

Запланований час виконання – 30 хв



## Завдання №2 VNS Lab 4 - Task 1 – Variant 5 :

Запланований час виконання – 15 хв

## Завдання №4 VNS Lab 5 - Task 1 – Variant 21 :

Запланований час виконання – 30 хв

## Завдання №4 Algotester Lab 2 - Variant 1 :

Запланований час виконання – 15 хв

## Завдання №5 Algotester Lab 3 - Variant 2 :

Запланований час виконання – 30 хв

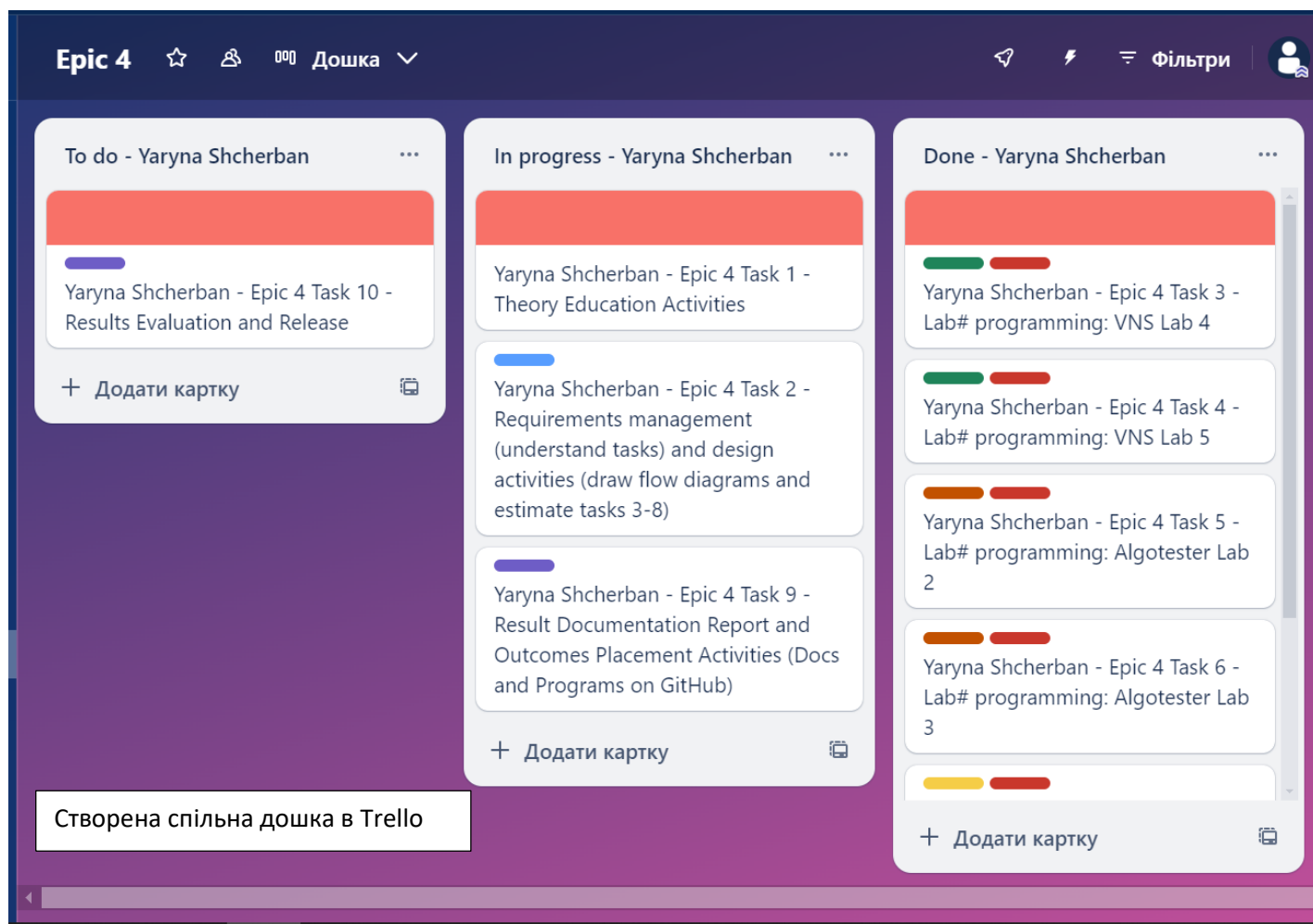
## Завдання №6 Algotester Lab 3 - Variant 3:

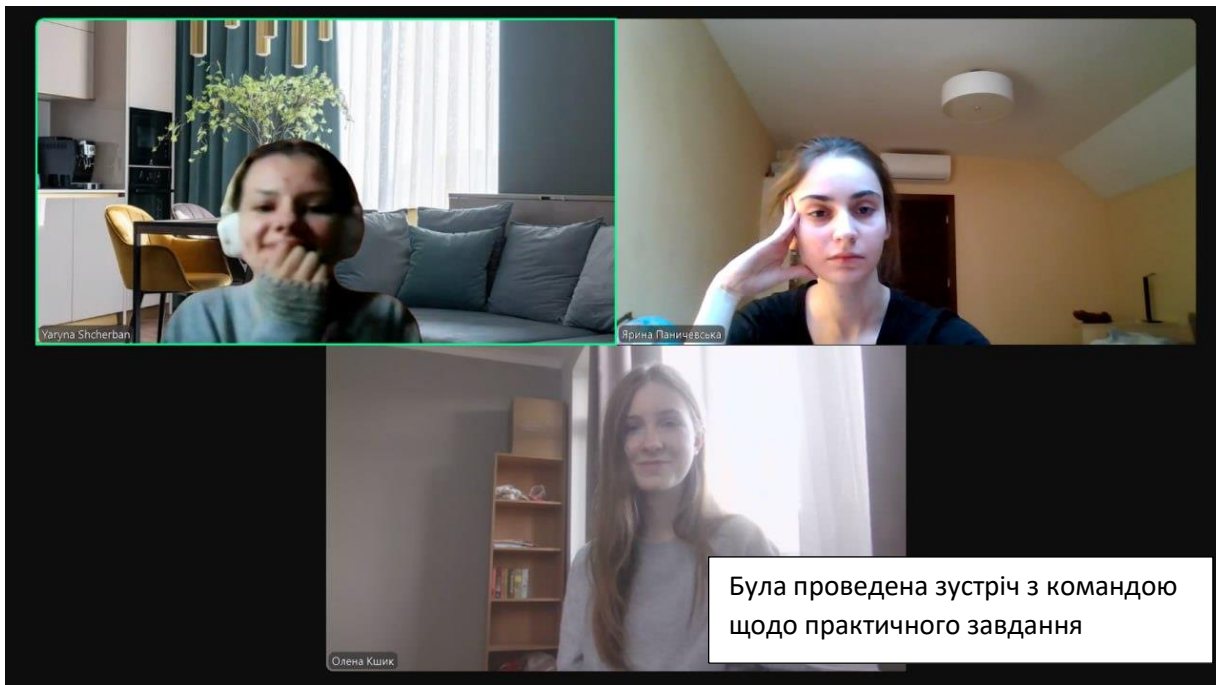
Запланований час виконання – 20 хв

## Завдання №7 Algotester : День програміста

Запланований час виконання – 20 хв

3) Конфігурація середовища до виконання завдань:





4) Код програми з посиланням на зовнішні ресурси

### Завдання №1 Епік 4 : Практичне завдання :

practice\_work\_task\_1\_yaryna\_shcherban.cpp

```
practice_work_task_1_yaryna_shcherban.cpp > main()
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
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 bool isPalindrome(int number) {
17     if (number < 0) {
18         return false;
19     }
20     string str = to_string(number);
21     return isPalindrome(str, 0, str.length() - 1);
22 }
23
24
25 int main() {
26
27     string word;
28     cout << "Enter a word to check if it is a palindrome: ";
29     cin >> word;
30 }
```

```

31     if (isPalindrome (word, 0, word.length() - 1) ) {
32         cout << "\"" << word << "\" is a palindrome." << endl;
33     } else {
34         cout << "\"" << word << "\" is not a palindrome." << endl;
35     }
36
37     int number;
38     cout << "Enter a number to check if it is a palindrome: ";
39     cin >> number;
40
41     if (isPalindrome(number)) {
42         cout << number << " is a palindrome." << endl;
43     } else {
44         cout << number << " is not a palindrome." << endl;
45     }
46
47     return 0;
48 }

```

## Завдання №2 VNS Lab 4 - Task 1 – Variant 5 :

vns\_lab\_4\_task\_variant\_5\_yaryna\_shcherban.cpp

```

vns_lab_4_task_variant_5_yaryna_shcherban.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include <cstdlib>
4  #include <ctime>
5
6  using namespace std;
7
8  int main() {
9
10     const int size = 10;
11     int Numbers[size];
12
13     srand(static_cast<unsigned int>(time(0)));
14     for (int i = 0; i < size; ++i) {
15         Numbers[i] = rand() % 100;
16     }
17
18     cout << "Array of random numbers \n";
19     for (int i = 0; i < size; ++i) {
20         cout << Numbers[i] << " ";
21     }
22     cout << endl;
23
24     vector<int> Numbers_2;
25     for (int i = 0; i < size; ++i) {
26         if (Numbers[i] % 7 != 0) {
27             Numbers_2.push_back(Numbers[i]);
28         }
29     }

```



```

31     vector<int> Numbers_3;
32     for (int num : Numbers_2) {
33         Numbers_3.push_back(num);
34         if (num % 2 != 0) {
35             Numbers_3.push_back(0);
36         }
37     }
38
39     cout << "Final array :\n";
40     for (int num : Numbers_3) {
41         cout << num << " ";
42     }
43     cout << endl;
44     return 0;
45 }

```

#### Завдання №4 VNS Lab 5 - Task 1 – Variant 21 :

vns\_lab\_5\_task\_variant\_21\_yaryna\_shcherban.cpp

```

G+ vns_lab_5_task_variant_21_yaryna_shcherban.cpp > findMaxRepeatedElement(const v
1  #include <iostream>
2  #include <vector>
3  #include <unordered_map>
4  #include <sstream>
5
6  using namespace std;
7
8  int findMaxRepeatedElement(const vector<vector<int>>& arr) {
9      unordered_map<int, int> frequencyMap;
10     int maxRepeated = -1;
11
12     for (const auto& row : arr) {
13         for (int num : row) {
14             frequencyMap[num]++;
15         }
16     }
17
18     //pair.second - частота , pair.first - значення
19
20     for (const auto& pair : frequencyMap) {
21         if (pair.second > 1) {
22             if (pair.first > maxRepeated) {
23                 maxRepeated = pair.first;
24             }
25         }
26     }
27     return maxRepeated;
28 }
29
30 int main() {

```



```

32     int rows, cols;
33     cout << "Enter the number of rows: ";
34     cin >> rows;
35     cout << "Enter the number of columns: ";
36     cin >> cols;
37     cin.ignore();
38
39     vector<vector<int>> array(rows, vector<int>(cols));
40
41     cout << "Enter the elements of the array :" << endl;
42     for (int i = 0; i < rows; i++) {
43         string line;
44         cout << "Row " << i + 1 << ": ";
45         getline(cin, line);
46
47         istringstream iss(line);
48         for (int j = 0; j < cols; j++) {
49             if (!(iss >> array[i][j])) {
50                 cout << "Invalid input. Please enter " << cols << " integers." << endl;
51                 return 1;
52             }
53         }
54     }
55
56     int maxRepeated = findMaxRepeatedElement(array);
57
58     if (maxRepeated != -1) {
59         cout << "Maximum of repeating elements : " << maxRepeated << endl;

```

```

60     } else {
61         cout << "There are no duplicate elements. " << endl;
62     }
63
64     return 0;
65 }

```

## Завдання №4 Algotester Lab 2 - Variant 1 :

algotester\_lab\_2\_task\_variant\_1\_yaryna\_shcherban.cpp

```
algotester_lab_2_task_variant_1_yaryna_shcherban.cpp > main()
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5  int main() {
6      int N;
7      cin >> N;
8
9      vector<int> r(N);
10     for (int i = 0; i < N; ++i) {
11         cin >> r[i];
12     }
13
14     sort(r.begin(), r.end());
15
16     if (r.size() == 1) {
17         cout << 0 << endl;
18         return 0;
19     }
20     int fatigue_1 = r[r.size() - 1] - r[1];
21     int fatigue_2 = r[r.size() - 2] - r[0];
22
23     if (fatigue_1 > fatigue_2) {
24         cout << fatigue_2 << endl;
25     } else {
26         cout << fatigue_1 << endl;
27     }
28     return 0;
29 }
```

## Завдання №5 Algotester Lab 3 - Variant 2 :

algotester\_lab\_3\_task\_variant\_2\_yaryna\_shcherban.cpp

```
algotester_lab_3_task_variant_2_yaryna_shcherban.cpp > main()
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <set>
5
6  using namespace std;
7
8  int main (){
9
10     int N;
11     int M;
12     vector <int> Vector1, Vector2;
13
14     cin >> N;
15     for (int i = 0; i < N; i++){
16         int Number;
17         cin >> Number;
18         Vector1.push_back(Number);
19     }
20
21     cin >> M;
22
23     for (int i = 0; i < M; i++){
24         int Number;
25         cin >> Number;
26         Vector2.push_back(Number);
27     }
28
29     int common = 0;
30     int unique = 0;
31
32     bool isCommon = false;
33     for (int i = 0; i < Vector1.size(); i++) {
34         for (int j = 0; j < Vector2.size(); j++) {
35             if (Vector1[i] == Vector2[j]) {
36                 common++;
37                 break;
38             }
39         }
40     }
41
42     unique = (Vector1.size() + Vector2.size()) - common;
43
44     cout << common << endl;
45     cout << unique << endl;
46
47     return 0;
48 }
```

### Завдання №6 Algotester Lab 3 - Variant 3:

self\_practice\_work\_algotester\_task\_1\_yaryna\_shcherban.cpp

```
self_practice_work_algotester_task_1_yaryna_shcherban.cpp > compressString(const strin
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  string compressString(const string& input) {
7      string compressed;
8      int length = input.length();
9
10     for (int i = 0; i < length; i++) {
11         char currentChar = input[i];
12         int count = 1;
13
14         while (i + 1 < length && input[i + 1] == currentChar) {
15             count++;
16             i++;
17         }
18         compressed += currentChar;
19         if (count > 1) {
20             compressed += to_string(count);
21         }
22     }
23     return compressed;
24 }
25
26 int main() {
27     string input;
28     cin >> input;
29
30     string compressed = compressString(input);
31     cout << compressed << endl;
32
33     return 0;
34 }
```

## Завдання №7 Algotester : День програміста

self\_practice\_work\_algotester\_task\_2\_yaryna\_shcherban.cpp

```
self_practice_work_algotester_task_2_yaryna_shcherban.cpp > 0
1  #include <iostream>
2  #include <string>
3  #include <set>
4  using namespace std;
5
6  int main() {
7      int n, m;
8      cin >> n >> m;
9      cin.ignore();
10
11     set<string> uniqueDrinks;
12     for (int i = 0; i < n; i++) {
13         string drink;
14         getline(cin, drink);
15         uniqueDrinks.insert(drink);
16     }
17
18     for (int i = 0; i < m; i++) {
19         string drink;
20         getline(cin, drink);
21         uniqueDrinks.insert(drink);
22     }
23
24     cout << uniqueDrinks.size() << endl;
25     return 0;
26 }
```

5) *Результати виконаних завдань, тестування та фактично затрачений час*

## Завдання №1 Епiк 4 : Практичне завдання :

Фактично затрачений час – 25 хв

```
Enter a word to check if it is a palindrome: level
"level" is a palindrome.
Enter a number to check if it is a palindrome: 123454321
123454321 is a palindrome.
PS C:\Yaryna\University_1\Programming\Epic 4> |
```

```
Enter a word to check if it is a palindrome: concious
"concious" is not a palindrome.
Enter a number to check if it is a palindrome: 1234213
1234213 is not a palindrome.
PS C:\Yaryna\University_1\Programming\Epic 4> |
```

## Завдання №2 VNS Lab 4 - Task 1 – Variant 5 :

Фактично затрачений час – 20 хв

```
Array of random numbers
91 24 47 22 22 59 8 63 63 17
Final array :
24 47 0 22 22 59 0 8 17 0
PS C:\Yaryna\University_1\Programming\Epic 4>

Array of random numbers
63 12 43 27 67 43 46 99 70 32
Final array :
12 43 0 27 0 67 0 43 0 46 99 0 32
PS C:\Yaryna\University_1\Programming\Epic 4>
```

## Завдання №4 VNS Lab 5 - Task 1 – Variant 21 :

Фактично затрачений час - 40 хв

```
Enter the number of rows: 5
Enter the number of columns: 6
Enter the elements of the array :
Row 1: 1 3 5 7 5 7
Row 2: 2 4 2 5 6 7
Row 3: 1 4 2 3 5 6
Row 4: 9 8 5 3 6 2
Row 5: 2 5 3 8 8 8
Maximum of repeating elements : 8
```

## Завдання №4 Algotester Lab 2 - Variant 1 :

Фактично затрачений час – 20 хв

a day ago	C++ 23	Accepted	0.003	1.195	<a href="#">View</a>
-----------	--------	----------	-------	-------	----------------------

```
5
1 2 2 4 4
2
PS C:\Yaryna\Un
5
1 2 2 2 4
1
PS C:\Yaryna\
```

## Завдання №5 Algotester Lab 3 - Variant 2 :

Фактично затрачений час – 30 хв

a day ago	C++ 23	Accepted	0.003	1.207	<a href="#">View</a>
-----------	--------	----------	-------	-------	----------------------

```
5
1 2 3 4 5
5
4 5 6 7 8
2
8
PS C:\Yaryna\University_1\Programming\Epic 4>
```

### Завдання №6 Algotester Lab 3 - Variant 3:

Фактично затрачений час – 15 хв

21 hours ago	C++ 23	Accepted	0.003	1.340	<a href="#">View</a>
<pre>AAAABVBCQQQ A4B3CQ4 PS C:\Yaryna\University_1\Programming\Epic 4&gt;</pre>					

### Завдання №7 Algotester : День програміста

Фактично затрачений час – 15 хв

21 hours ago	<a href="#">0211 - День програміста</a>	C++ 23	Accepted	0.003	1.168
<pre>1 2 medoff tuborg medoff 2 PS C:\Yaryna\University_1\Programming\Epic 4&gt;</pre>					

Висновок : На цій лабораторній роботі я опрацювала створення масивів різних вимірів, роботу з вказівниками та посиланнями, а також класи пам'яті в C++; дослідила алгоритми сортування та їхнє використання для різних випадків; навчилася створювати структури, що допомогло об'єднати різні типи даних для зручнішої роботи.

Посилання на Pull Request : [https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/297](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/297)