

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 ІІІ-11

Федоришин Микола Володимирович

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:

Закріпити на практиці вивчений теоретичний матеріал, зокрема робота з масивами, двовимірними та одновимірними, а також з вказівниками та посиланнями. Також розібратися з динамічними масивами та динамічною пам'яттю в C++.

Теоретичні відомості:

- Тема №1: Класи пам'яті в C++.
- Тема №2: Вступ до Масивів і Вказівників.
- Тема №3: Одновимірні Масиви.
- Тема №4: Вказівники та Посилання.
- Тема №5: Двовимірні Масиви.
- Тема №6: Динамічні Масиви.
- Тема №7: Структури Даних.
- Тема №8: Вкладені Структури.
- Тема №9: Використання Структур.
- Тема №10: Алгоритми обробки та робота з Масивами та

структурами.

1) Індивідуальний план опрацювання теорії:

- Тема №1: Класи пам'яті в C++:
 - o Джерела інформації:
 - Статті.
<http://cpp.dp.ua/klasy-pam-yati-u-c-builder/>
 - Що опрацьовано:
 - o Статична пам'ять.
 - o Динамічна пам'ять.
 - o Поняття стеку.
 - o Виділення та вивільнення пам'яті.
- Запланований час на вивчення 30 хвилин.
Витрачений час 30 хвилин.
- Тема №2: Вступ до Масивів і Вказівників:
 - o Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=ULdbOaMBPYc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=40>
<https://www.youtube.com/watch?v=zopWRIYOWw&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=57>
 - Що опрацьовано:
 - o Основи масивів: визначення, важливість, приклади використання.
 - o Різниця між статичними та динамічними масивами.
 - o Основи вказівників: що це таке, як вони працюють.
 - o Взаємозв'язок між масивами та вказівниками.
 - o Вступ до посилань: основні концепції та відмінності від вказівників.
- Запланований час на вивчення 1 година.
Витрачений час 1 година.

- Тема №3: Одновимірні масиви:

- Джерела інформації:
 - Статті.

<https://www.youtube.com/watch?v=ULdbOaMBPYc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=40>

- Що опрацьовано

- Створення та ініціалізація одновимірних масивів.
- Основні операції: індексація, присвоєння, читання.
- Цикли та обхід масивів.
- Використання функцій для роботи з масивами.
- Приклади алгоритмів сортування та пошуку.

Запланований час на вивчення 1 година.

Витрачений час 1 година.

- Тема №4: Вказівники та Посилання:

- Джерела інформації:
 - Статті.

<https://www.youtube.com/watch?v=zopWRIYOWw&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=57>

- Що опрацьовано

- Використання вказівників для доступу до елементів масиву.
- Арифметика вказівників.
- Різниця між вказівниками та посиланнями в контексті функцій.
- Динамічне виділення пам'яті з використанням вказівників.
- Використання вказівників для створення складних структур даних.

Запланований час на вивчення 2 години.

Витрачений час 2 години.

- Тема № 5: Двовимірні Масиви:

- Джерела інформації:
 - Статті.

<https://www.youtube.com/watch?v=hcYgFCgeZzQ>

- Що опрацьовано

- Оголошення та ініціалізація двовимірних масивів.
- Вкладені цикли для обходу двовимірних масивів.
- Практичні приклади використання двовимірних масивів.
- Передача двовимірних масивів у функції.
- Застосування двовимірних масивів для розв'язання задач.

Запланований час на вивчення 1 година.

Витрачений час 1 година..

- Тема №6: Динамічні Масиви:

- Джерела інформації:
 - Статті.

<https://www.youtube.com/watch?v=OGR9VJEh8Hk&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=60>

https://www.youtube.com/watch?v=_N3zkbnCTw0

- Що опрацьовано

- Основи динамічного виділення пам'яті.
- Створення та управління динамічними масивами.
- Використання операторів new та delete для управління пам'яттю.
- Реалізація змінної розмірності масивів.

- Передача динамічних масивів у функції.
 - Запланований час на вивчення 1 година.
 - Витрачений час 1 година.
- Тема №7: Структури Даних:
 - Джерела інформації:
 - Статті.
 - https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63
 - https://www.youtube.com/watch?v=D79J_q5SGzc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=64
 - <https://www.youtube.com/watch?v=ifdk1fYSk1Y&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=65>
 - Що опрацьовано
 - Оголошення та використання структур.
 - Використання масивів та вказівників у структурах.
 - Функції для обробки даних у структурах.
 - Використання структур для представлення складних даних.
 - Вкладені структури та їх використання.
 - Об'єднання (Union)
 - Переліки (enumerations)
 - Запланований час на вивчення 2.5 години.
 - Витрачений час 2.5 години.
- Тема №8: Вкладені Структури:
 - Джерела інформації:
 - Статті.
 - https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63
 - Що опрацьовано
 - Поняття вкладених структур та їх оголошення.
 - Взаємодія з вкладеними структурами.
 - Використання вкладених структур для моделювання складних даних.
 - Передача вкладених структур у функції.
 - Приклади реального використання вкладених структур.
 - Запланований час на вивчення 1 година.
 - Витрачений час 1 година.
- Тема №9: Використання Структур:
 - Джерела інформації:
 - Статті.
 - https://www.youtube.com/watch?v=999IE-6b7_s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=63
 - Що опрацьовано
 - Перевантаження операторів у структурі.
 - Вивід/ввід структури (operator<<);
 - Арифметичні операції з структурами (operator+, operator-);
 - Практичні задачі на виведення структур та операції з ними
 - Запланований час на вивчення 1 година.
 - Витрачений час 1 година.
- Тема №10: Алгоритми обробки та робота з Масивами та структурами:
 - Джерела інформації:
 - Статті.

<https://www.youtube.com/watch?v=uQxG9gBROog&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=51>

<https://www.youtube.com/watch?v=maB87eyn7h8&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=52>

- Що опрацьовано
 - o Алгоритми пошуку та сортування в масивах.
 - o Обробка та маніпуляції з даними у структурах.
 - o Використання циклів та умовних операторів для роботи з масивами та структурами.
 - o Інтеграція масивів та структур у алгоритми.
 - o Розв'язання практичних задач з використанням масивів та структур
- Запланований час на вивчення 1.5 години.
Витрачений час 1.5 години.

Виконання роботи:

1. Опрацювання завдання та вимог до програм.

Завдання №1

VNS LAB 4 – TASK 1 (VARIANT14)

1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел. 2) Роздрукувати отриманий масив. 3) Перевернути масив і, якщо кількість елементів масиву непарна, знищити його середній елемент. 4) Додати на початок масиву 3 елементи зі значенням $M[I+10]-2$. 5) Роздрукувати отриманий масив.

Завдання №2

VNS LAB 5 – TASK 1 (VARIANT 14)

Обчислити добуток всіх стовпців масиву, у яких перший елемент більший від елементів розташованих на головній і бічній діагоналі.

Завдання №3

ALGOTESTER LAB 2 (VARIANT 1)

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?

Завдання №4

ALGOTESTER LAB 3 (VARIANT 2)

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.

Завдання №5

CLASS PRACTICE WORK

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

допомогою рекурсії.

Вимоги

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

Завдання №6

SELF PRACTICE WORK ALGOTESTER

Сьогодні в пластунів свято! Привезли вдвічі більше продуктів на обід. Зеник разом із Марічкою зголосилися допомагати на кухні. Всього є три різні страви: борщ, картопля і салат.

У Пласті в Зеника та Марічки є n друзів. i -й друг хоче з'їсти a_i грамів борщу, b_i грамів картоплі та c_i грамів салату. Якщо друг Зеника та Марічки хоче з'їсти x грамів певної страви, а йому в тарілку поклали y грамів, то ступінь недовіри до Зеника з Марічкою в цього друга зростає на $|x - y|$ одиниць. Зенику з Марічкою наказали накладати всім однакові порції певної страви. Тобто кожному другу потрібно накласти A грамів борщу, B грамів картоплі та C грамів салату. Допоможіть Зенику та Марічці обрати такі A , B та C , щоб сумарний ступінь недовіри всіх друзів був якомога меншим.

Input

У першому рядку задано одне ціле число n — кількість друзів Зеника та Марічки. У наступних n рядках задано по три цілі числа a_i , b_i , c_i — кількість борщу, картоплі та салату, яку хоче отримати i -ий друг.

Output

У єдиному рядку виведіть одне ціле число — сумарний ступінь недовіри всіх друзів до Зеника та Марічки.

2. Дизайн та планувальна оцінка часу виконання

завдань:

Програма №1

Плановий час на реалізацію 1.5 години.

Програма №2

Плановий час на реалізацію 2 години.

Програма №3

Плановий час на реалізацію 1 година.

Програма №4

Плановий час на реалізацію 1.5 години.

Програма №5

Блок – схема.

Плановий час на реалізацію 1 година.

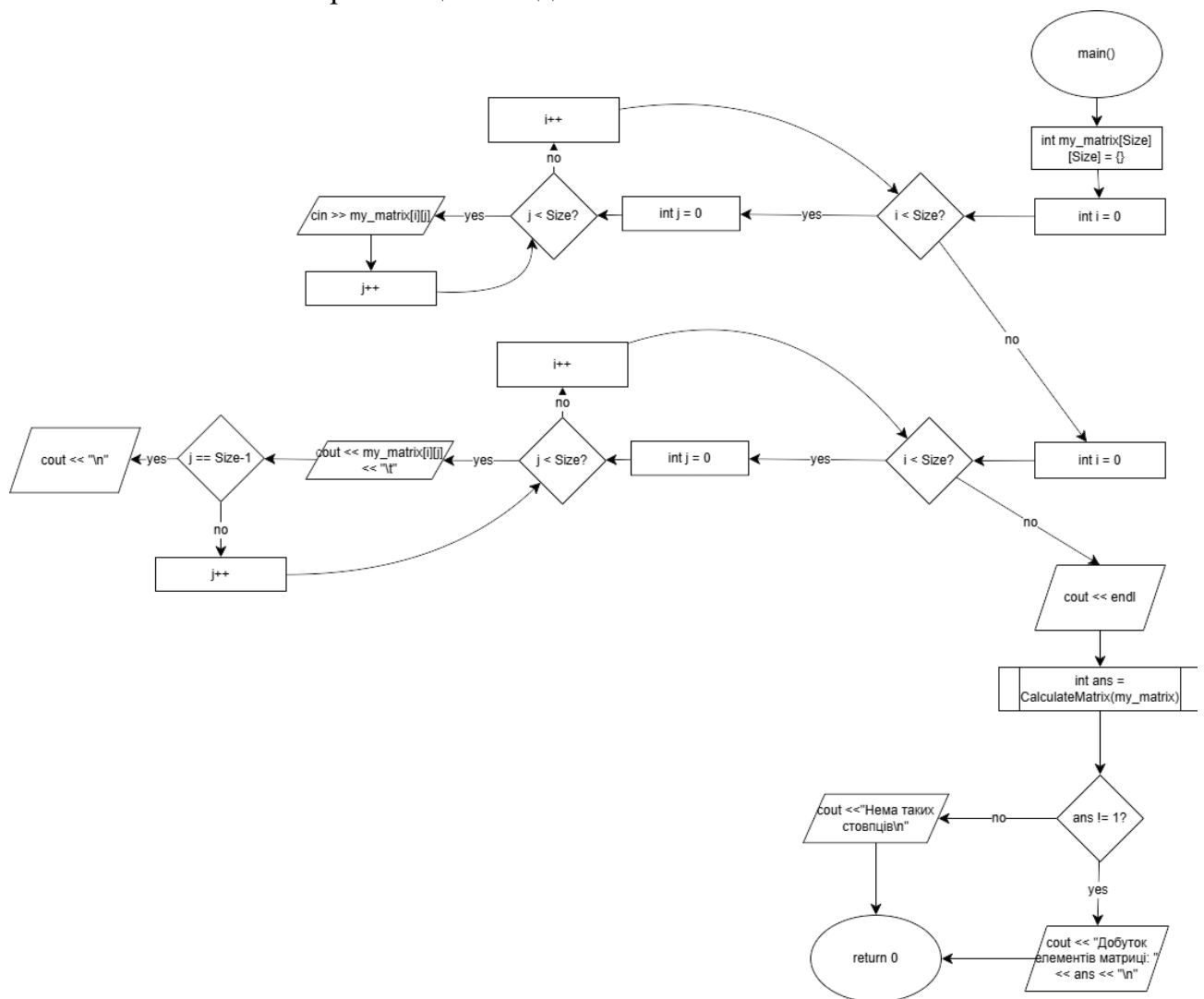


Рисунок 2.1. Блок-схема до функції `main`

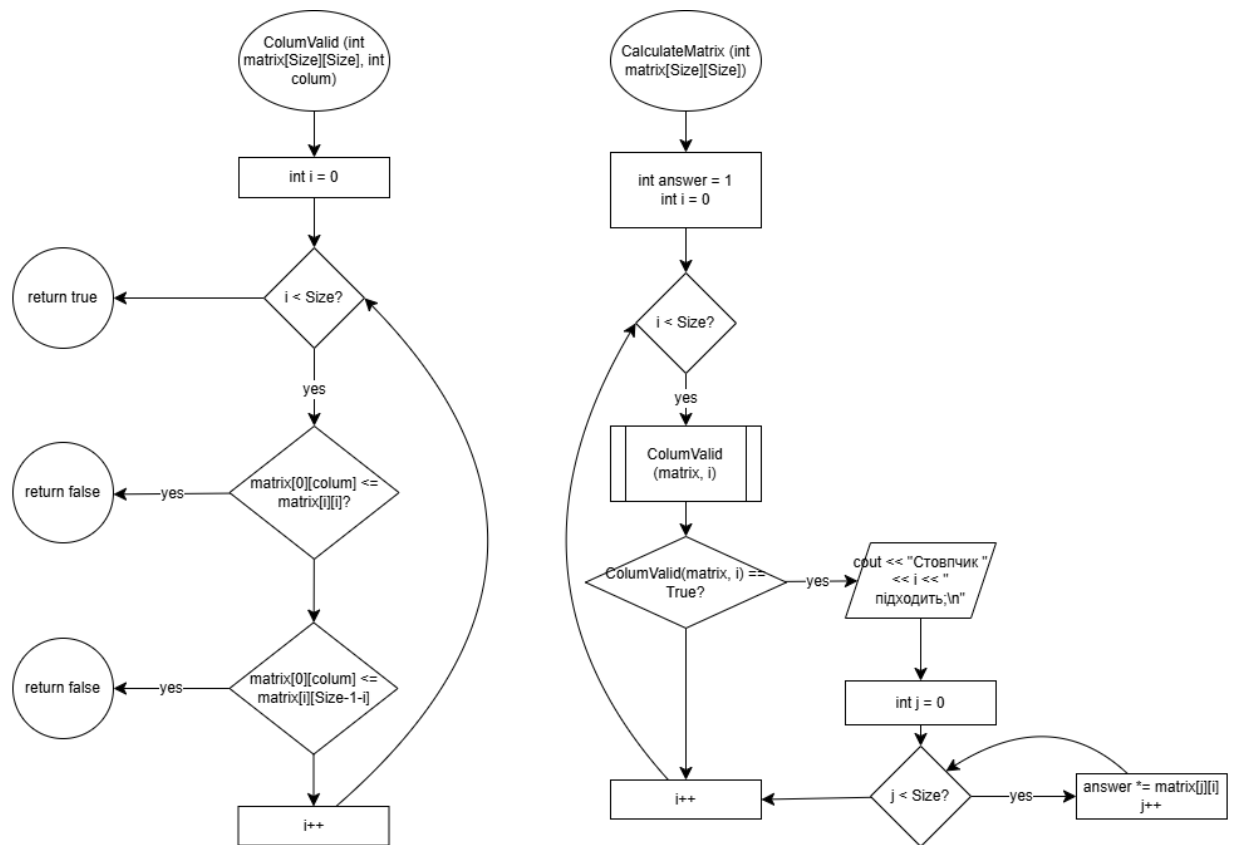


Рисунок 2.2. Блок-схема до функцій які викликаю у main

Програма №6

Плановий час на реалізацію 2 години.

3. Код програм з посиланням на зовнішні ресурси та фактично затрачений час:

Завдання №1

```
G:\vns_lab_4_task_1_variant_1_mykola_fedoryshyn.cpp U X
lab_4 > G:\vns_lab_4_task_1_variant_1_mykola_fedoryshyn.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      const int N = 103;
6      int a[N];
7      int reallength;
8
9      cout << "Введіть реальну довжину масиву (не більше 100): ";
10     cin >> reallength;
11     if (reallength > N) {
12         cout << "Задана довжина перевищує максимальну, завершення програми.\n";
13         return 0;
14     }
15
16     srand(time(0));
17     cout << "Початковий масив: ";
18     for (int i = 0; i < reallength; i++) {
19         a[i] = rand() % 101;
20         cout << a[i] << " ";
21     }
22     cout << endl;
23
24     /* Крок 3: Перевертаємо масив */
25     for (int i = 0; i < reallength / 2; i++) {
26         int temp = a[i];
27         a[i] = a[reallength - i - 1];
28         a[reallength - i - 1] = temp;
29     }
30
31     /* Якщо кількість елементів непарна, видаляємо середній елемент */
32     if (reallength % 2 != 0) {
33         int mid = reallength / 2;
34         for (int i = mid; i < reallength - 1; i++) {
35             a[i] = a[i + 1];
36         }
37         reallength--;
38     }
39
40     if (reallength >= 10) {
41         /* Зміщення елементів на 3 вправо */
42         for (int i = reallength - 1; i >= 0; i--) {
43             a[i + 3] = a[i];
44         }
45         for (int i = 0; i < 3; i++) {
46             a[i] = a[i + 10] - 2;
47         }
48         reallength += 3;
49     } else {
50         cout << "Недостатня кількість елементів для обчислення за формулою a[i + 10] - 2. Мінімальна кількість це 10.\n";
51     }
52
53     cout << "Отриманий масив: ";
54     for (int i = 0; i < reallength; i++) {
55         cout << a[i] << " ";
56     }
57     return 0;
58 }
```

Рисунок 3.1. Код до програми № 1

```
Введіть реальну довжину масиву (не більше 100): 27
Початковий масив: 79 82 42 12 52 21 14 60 91 48 31 79 18 37 90 21 4 84 54 17 67 13 59 64 47 40 6
Отриманий масив: 15 52 82 6 40 47 64 59 13 67 17 54 84 4 21 90 18 79 31 48 91 60 14 21 52 12 42 82 79
```

Рисунок 3.2. Приклад виконання програми № 1

Посилання на файл у пул реквесті

Завдання №2

```
G:\vns_lab_5_task_1_variant_14_mykola_fedoryshyn.cpp U
lab_5 > G:\vns_lab_5_task_1_variant_14_mykola_fedoryshyn.cpp > ...
1  #include <iostream>
2  using namespace std;
3  const int Size = 4;
4  int CalculateMatrix(int matrix[Size][Size]);
5  bool ColumValid(int matrix[Size][Size], int colum);
6
7  int main(){
8      int my_matrix[Size][Size] = {};
9      for(int i = 0; i<Size; i++){
10         for(int j = 0; j<Size; j++){
11             cin >> my_matrix[i][j];
12         }
13     }
14     for(int i = 0; i<Size; i++){
15         for(int j = 0; j<Size; j++){
16             cout << my_matrix[i][j] << "\t";
17             if(j == Size-1){
18                 cout << "\n";
19             }
20         }
21     }
22     cout << endl;
23     int ans = CalculateMatrix(my_matrix);
24     if(ans != 1){
25         cout << "Добуток елементів матриці: " << ans << "\n";
26     }else{
27         cout << "Нема таких стовпців\n";
28     }
29     return 0;
30 }
31
32
33 bool ColumValid(int matrix[Size][Size], int colum){
34     for(int i = 0; i<Size; i++){
35         if(matrix[0][colum] <= matrix[i][i]){
36             return false;
37         }
38         if(matrix[0][colum] <= matrix[i][Size-1-i]){
39             return false;
40         }
41     }
42     return true;
43 }
44 int CalculateMatrix(int matrix[Size][Size]){
45     int answer = 1;
46     for(int i = 0; i<Size; i++){
47         if (ColumValid(matrix, i)){
48             cout << "Стовпчик " << i << " підходить;\n";
49             for(int j = 0; j<Size; j++){
50                 answer *= matrix[j][i];
51             }
52         }
53     }
54     return answer;
55 }
```

Рисунок 3.3. Код до програми № 2

```
1 10 8 2
1 2 3 4
4 3 2 1
6 5 2 4

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

Стовпчик 1 підходить;
Стовпчик 2 підходить;
Добуток елементів матриці: 28800
```

Рисунок 3.4. Приклад виконання програми № 2

Завдання №3

```
algotester_lab_2 > G+ algotester_lab_2_variant_1_mykola_fedoryshyn.cpp > main()
1  #include <iostream>
2  using namespace std;
3  int main(){
4      int size;
5      bool max = true, min = true;
6
7      std::cin >> size;
8      int list[size];
9      for(int i = 0; i < size; i++){
10         std::cin >> list[i];
11     }
12     /*Якщо масив має 1 значення то вивід завжди 0*/
13     if(size == 1 || size == 0){
14         cout << 0;
15         return 0;
16     }
17     for(int i = 1; i < size; i++){
18         int j = i-1, index = list[i];
19         for(j; j > -1 && list[j] > index; j--){
20             list[j+1] = list[j];
21         }
22         list[j+1] = index;
23     }
24
25     /*Перевірка чи максимальний і мінімальний елементи не повторюються*/
26     if(list[size-1] == list[size-2]){
27         max = false;
28     }
29     else if(list[0] == list[1]){
30         min = false;
31     }
32
33     if(max && min){
34         if (list[size-2] - list[0] > list[size-1] - list[1]){
35             cout << list[size-1] - list[1];
36         }
37         else{
38             cout << list[size-2] - list[0];
39         }
40     }
41     else if(max){
42         cout << list[size-2] - list[0];
43     }
44     else if(min){
45         cout << list[size-1] - list[1];
46     }
47     else{
48         cout << list[size-1] - list[0];
49     }
50 }
```

Рисунок 3.5. Код до програми №3

```
5
1 3 3 4 4
1
```

Рисунок 3.6. Приклад виконання програми №3

a day ago	C++ 20	Accepted	0.003	1.063	View
-----------	--------	----------	-------	-------	----------------------

Рисунок 3.7. Статус задачі на алготестері

Фактично затрачений час 1 година.

Посилання на файл у пулл реквесті

Завдання №4

```
algotester_lab_3 > g++ algotester_lab_3_variant_2_mykola_fedoryshyn.cpp > main0
1  #include "iostream"
2  #include "string"
3  #include "cmath"
4  using namespace std;
5
6  int main(){
7      int size1, size2, n;
8
9      cin >> size1;
10     int list1[size1];
11     for(int i = 0; i < size1; i++){
12         cin >> list1[i];
13     }
14     cin >> size2;
15     int list2[size2];
16     for(int i = 0; i < size2; i++){
17         cin >> list2[i];
18     }
19
20     for(int i = 0; i < size1; i++){
21         for(int j = 0; j < size2; j++){
22             if(list1[i] == list2[j]){
23                 n++;
24             }
25         }
26     }
27     cout << n << endl << size1+size2 - n;
28     return 0;
29 }
```

Рисунок 3.8. Код до програми №4

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

Рисунок 3.9. Приклад виконання програми №4

a day ago	C++ 20	Accepted	0.003	1.227	View
-----------	--------	----------	-------	-------	----------------------

Рисунок 3.10. Статус задачі на алготестері

Фактично затрачений час 10хв.

Посилання на файл у пулл реквесті

Завдання №5

```
practice > G+ practice_work_task_1_mykola_fedoryshyn.cpp > main()
1  #include <iostream>
2  #include <string>
3  #include <cmath>
4  using namespace std;
5
6
7  bool isPalindrome(const string& str, int start, int end) {
8      if (start >= end) {
9          return true;
10     }
11     if (str[start] != str[end]) {
12         return false;
13     }
14     return isPalindrome(str, start + 1, end - 1);
15 }
16 bool isPalindrome(int num) {
17     if (num < 0){return false;}
18
19     int reversed = 0, original = num;
20     while (num != 0) {
21         int digit = num % 10;
22         reversed = reversed * 10 + digit;
23         num /= 10;
24     }
25     return original == reversed;
26 }
27
28
29 int main() {
30     string str;
31     cout << "Введіть рядок: ";
32     cin >> str;
33     if (isPalindrome(str, 0, str.length() - 1)) {
34         cout << "Рядок є паліндромом.\n";
35     } else {
36         cout << "Рядок не є паліндромом.\n";
37     }
38
39
40     int num;
41     cout << "Введіть число: ";
42     cin >> num;
43     if (isPalindrome(num)) {
44         cout << "Число є паліндромом.\n";
45     } else {
46         cout << "Число не є паліндромом.\n";
47     }
48
49     return 0;
50 }
```

Рисунок 3.11. Код до програми №5

```
Введіть рядок: qwewq
Рядок є паліндромом.
Введіть число: 12312
Число не є паліндромом.
```

Рисунок 3.12. Приклади виконання програми №5

Фактично затрачений час 2 години.

Посилання на файл у пулл реквесті

Завдання №6

```
self_practice > G self_practice_work_algotester_task_1_mykola_fedoryshyn.cpp > main()
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <cmath>
5  using namespace std;
6
7
8  int Median(vector<int>& arr) {
9      sort(arr.begin(), arr.end());
10     int n = arr.size();
11     if (n % 2 == 1) {
12         return arr[n/2];
13     } else {
14         return (arr[n/2 - 1] + arr[n / 2]) / 2;
15     }
16 }
17
18 int main() {
19     int n;
20     cin >> n;
21
22     vector<int> borsch(n), potato(n), salad(n);
23
24     for (int i = 0; i < n; i++) {
25         cin >> borsch[i] >> potato[i] >> salad[i];
26     }
27     int A = Median(borsch);
28     int B = Median(potato);
29     int C = Median(salad);
30
31     int ans = 0;
32     for (int i = 0; i < n; i++) {
33         ans += abs(borsch[i] - A);
34         ans += abs(potato[i] - B);
35         ans += abs(salad[i] - C);
36     }
37
38     cout << ans << endl;
39     return 0;
40 }
```

Рисунок 3.13. Код до програми №6

```
2
10 50 100
70 10 50
150
```

Рисунок 3.14. Приклад виконання програми №6

an hour ago	C++ 20	Accepted	0.098	2.285	View
-------------	--------	----------	-------	-------	------

Рисунок 3.15. Статус задачі на алготестері

Фактично затрачений час 1.5 години.

Посилання на файл у пулл реквесті

4. Робота з командою:

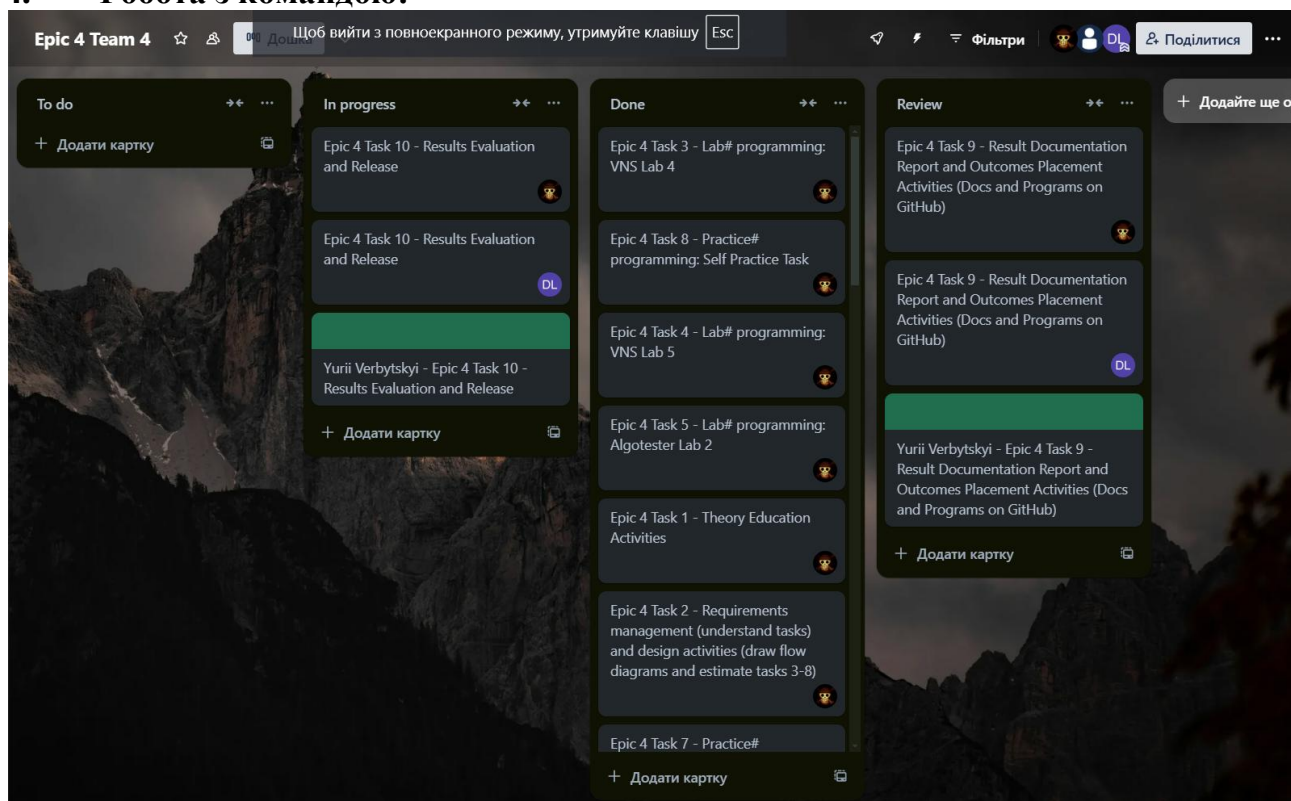


Рисунок 4.1. Командна дошка в Trello

Висновок: У межах практичних та лабораторних робіт блоку №4, я вивчив багато нового матеріалу, такого як: різні масиви одновимірні та двовимірні, вказівники та посилання, динамічні масиви та структури даних. Застосувавши вивчений матеріал на практиці краще зрозумів як це все працює і як це реалізовувати. Також створив блок-схему до найважчої задачі, і краще зрозумів як працює програма. Створив дошку в Trello для роботи з командою.