



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 ШІ-11

Єдинець Євген Русланович

Тема роботи:

Вивчення роботи з даними на мові C++: одновимірні та двовимірні масиви, вказівники, посилання, динамічні масиви, структури даних, вкладені структури, алгоритми обробки масивів та структур.

Мета роботи:

Ознайомитися з роботою з даними в C++, зокрема з одновимірними та двовимірними масивами, вказівниками та посиланнями. Набути навичок створення та використання динамічних масивів і структур даних, включаючи вкладені структури. Вивчити алгоритми обробки масивів та структур для створення ефективних і зрозумілих програм.

Теоретичні відомості:

1. Одновимірні масиви

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

2. Багатовимірні масиви

<https://www.youtube.com/watch?v=V2g3B9Zbh4Q&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=41>

3. Вказівники та посилання

<https://www.youtube.com/watch?v=s7H2ScLcH88&list=PL7vq4D0vOpQa9WaLe7btV01eixBUZ6-Ve&index=13>

4. Динамічні масиви

<https://www.youtube.com/watch?v=qUX4xCRB8FM>

5. Структури даних

<https://www.youtube.com/watch?v=dB2V9f0R9uk&t=3s>

Виконання роботи:

1) Опрацювання завдання та вимог до програм та середовища:

Завдання 1

VNS Lab 4

- 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 K -ого елемента і до $K-1$ (по кільцю вліво).
- 3) Додати в кільце перший і 3 останніх елементи.
- 4) Роздрукувати отриманий масив, починаючи з K -ого елемента (і до $K+1$ по кільцю вправо).

Завдання 2

VNS Lab 5

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

Знайти максимальний з повторюваних елементів двовимірного масиву.

Завдання 3

Algotester Lab 2

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135593>

Завдання 4

Algotester Lab 3

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135596>

Завдання 5

Class Practice Work

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.

Завдання 6

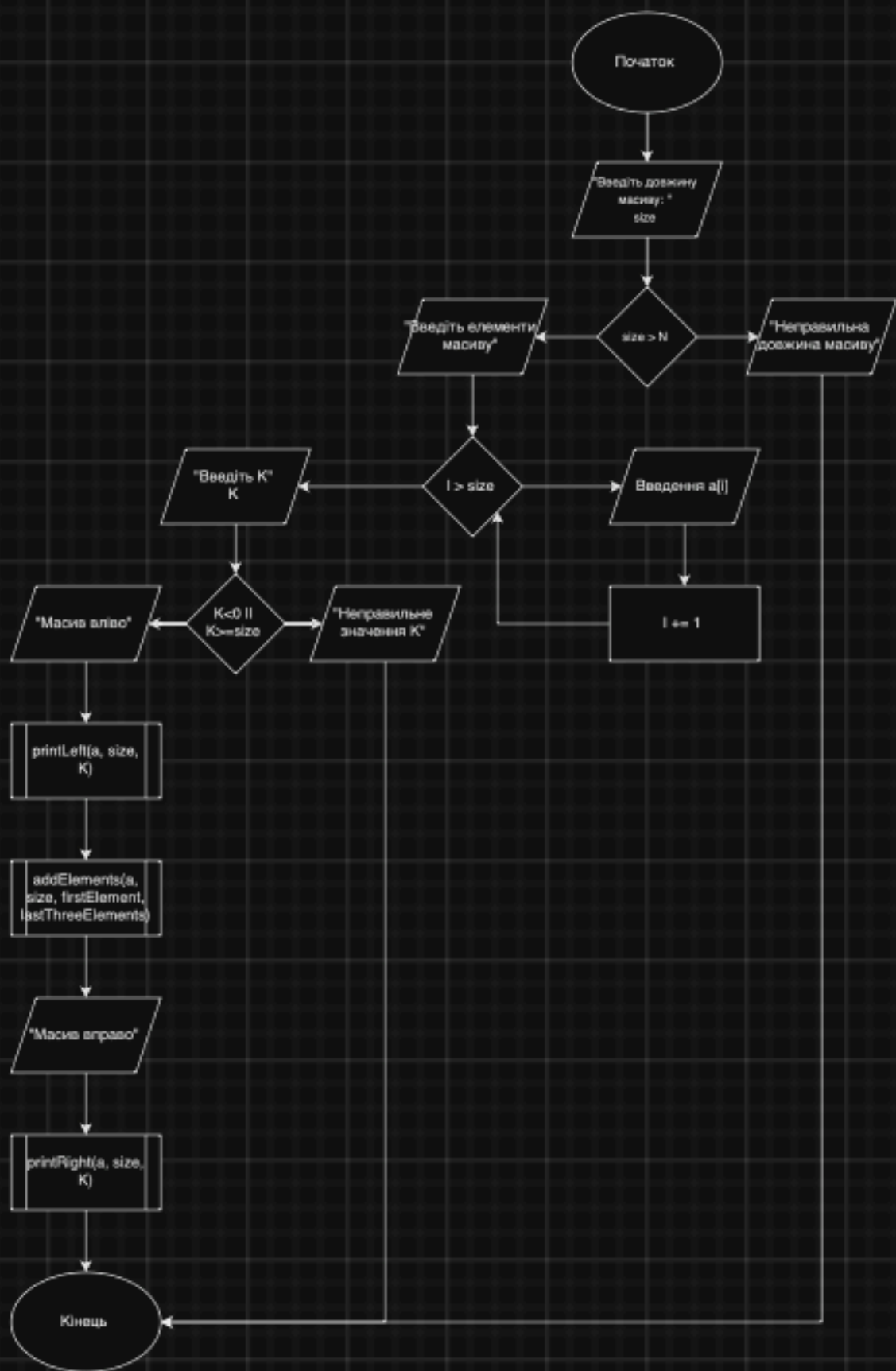
Self Practice Work

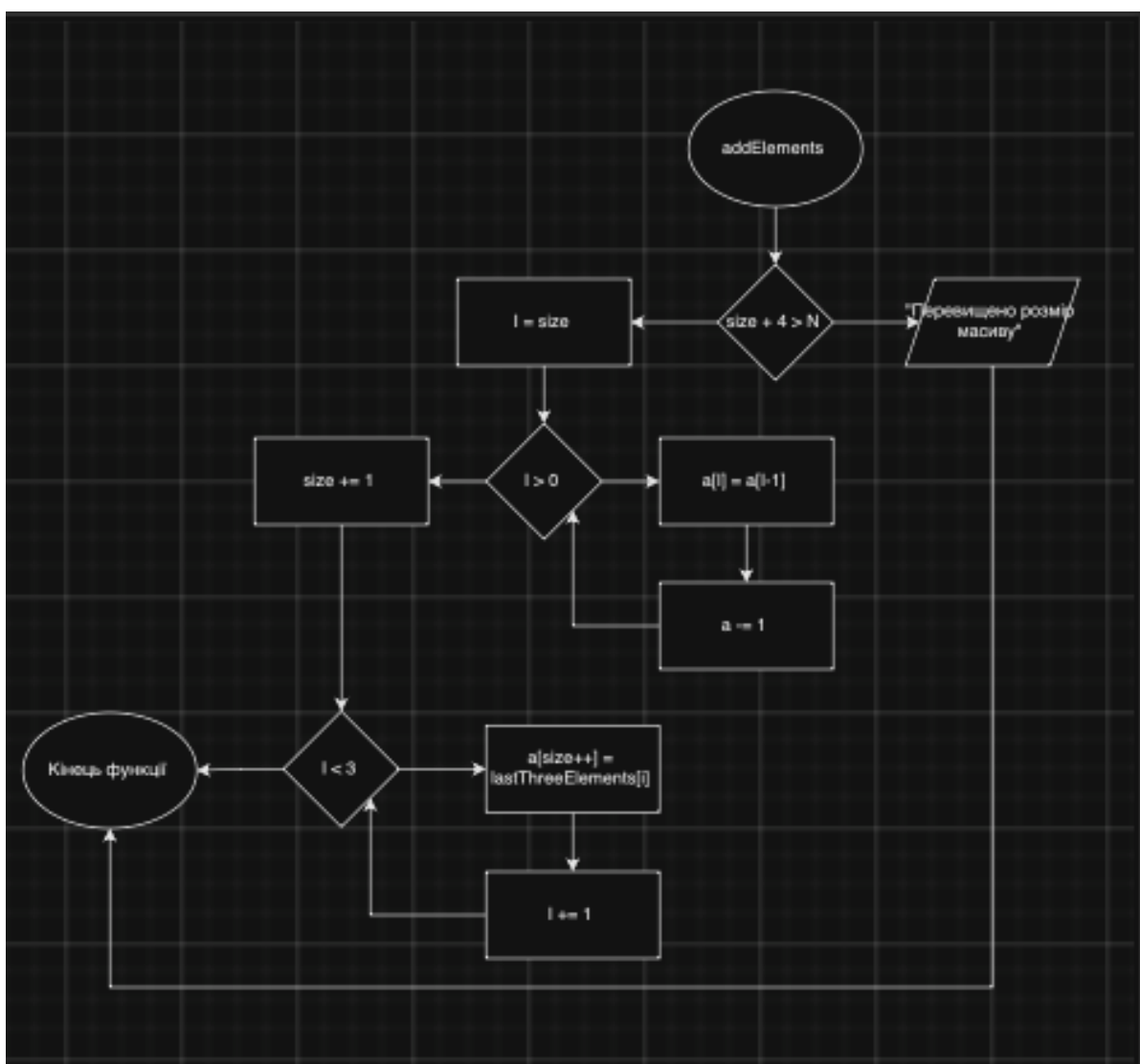
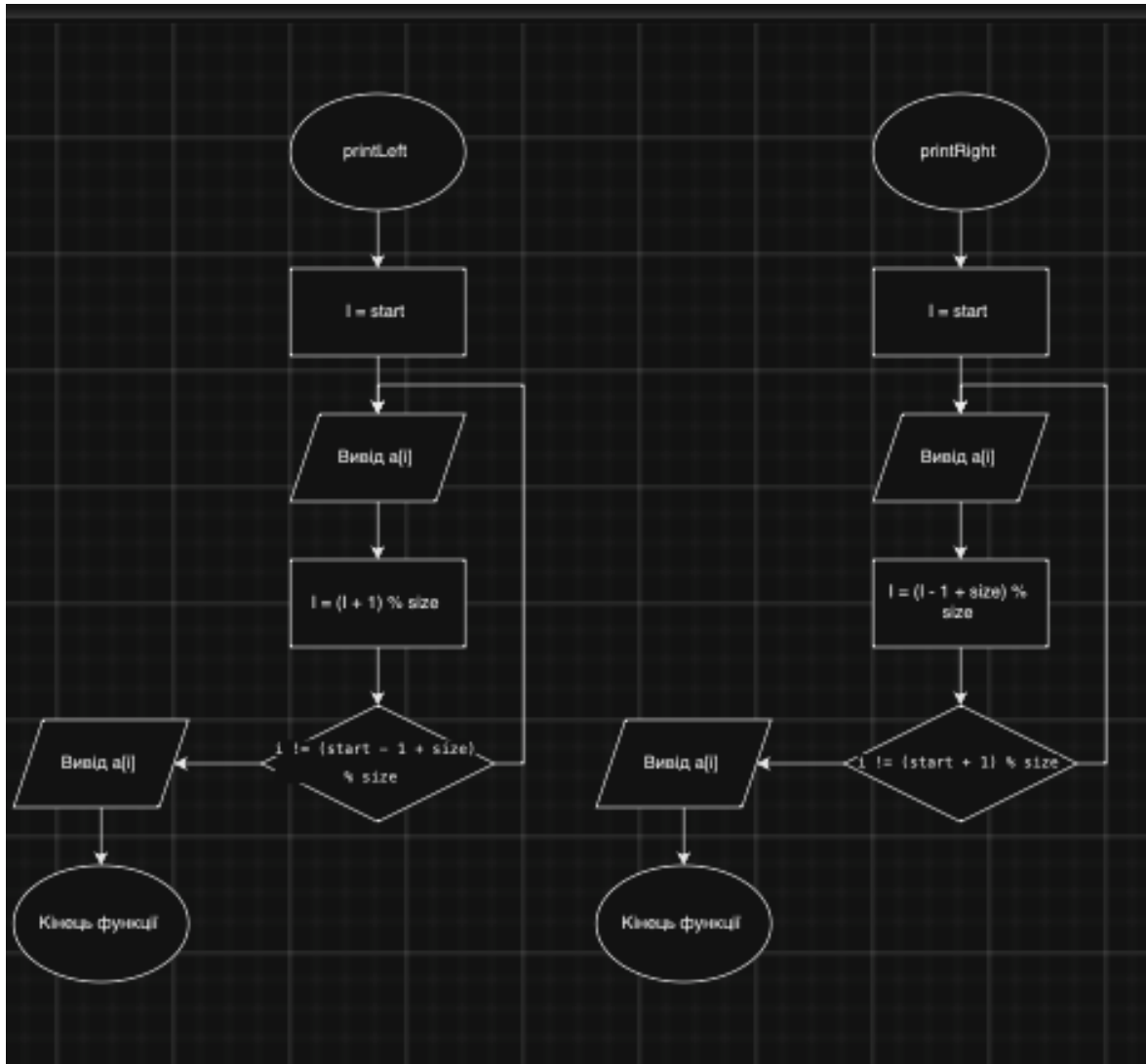
Веселі каруселі

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40265#mySolutions>

2) Дизайн та планована оцінка часу виконання завдань:

Завдання 1





3) Код програм:

Завдання 1:

```
1  #include <iostream>
2
3  using namespace std;
4
5
6  const int N = 100;
7
8  void printLeft(int a[], int size, int start) {
9      int i = start;
10     do {
11         cout << a[i] << " ";
12         i = (i + 1) % size;
13     } while (i != (start - 1 + size) % size);
14     cout << a[i] << endl;
15 }
16
17 void printRight(int a[], int size, int start) {
18     int i = start;
19     do {
20         cout << a[i] << " ";
21         i = (i - 1 + size) % size;
22     } while (i != (start + 1) % size);
23     cout << a[i] << endl;
24 }
25
26 void addElements(int a[], int& size, int firstElement, int lastThreeElements[]) {
27     if (size + 4 > N) {
28         cout << "Перевищено максимальний розмір масиву" << endl;
29         return;
30     }
31
32     for (int i = size; i > 0; --i) {
33         a[i] = a[i - 1];
34     }
35     a[0] = firstElement;
36     ++size;
37
38     for (int i = 0; i < 3; ++i) {
39         a[size++] = lastThreeElements[i];
40     }
41 }
```



```

42
43 int main() {
44     int a[N];
45     int size;
46
47     cout << "Введіть довжину масиву: ";
48     cin >> size;
49
50     if (size > N) {
51         cout << "Неправильна довжина масиву!" << endl;
52         return 1;
53     }
54
55     cout << "Введіть елементи масиву: ";
56     for (int i = 0; i < size; ++i) {
57         cin >> a[i];
58     }
59
60     int K;
61     cout << "Введіть номер K (від 0 до " << size - 1 << "): ";
62     cin >> K;
63
64     if (K < 0 || K >= size) {
65         cout << "Неправильне значення K!" << endl;
66         return 1;
67     }
68
69     cout << "Масив вліво від K до K-1: ";
70     printLeft(a, size, K);
71
72     int firstElement = a[0];
73     int lastThreeElements[] = {a[size - 3], a[size - 2], a[size - 1]};
74     addElements(a, size, firstElement, lastThreeElements);
75
76     cout << "Масив вправо від K до K+1: ";
77     printRight(a, size, K);
78
79     return 0;
80 }
81

```

Завдання 2:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int findMaxRepeated(int arr[3][3], int rows, int cols) {
6      int freq[100] = {0};
7
8      for (int i = 0; i < rows; ++i) {
9          for (int j = 0; j < cols; ++j) {
10             freq[arr[i][j]]++;
11         }
12     }
13
14     int maxRepeated = -1;
15     int maxCount = 1;
16
17     for (int i = 0; i < 100; ++i) {
18         if (freq[i] > maxCount) {
19             maxCount = freq[i];
20             maxRepeated = i;
21         }
22     }
23
24     return maxRepeated;
25 }
26
27 int main() {
28     int arr[3][3] = {
29         {1, 2, 3},
30         {3, 2, 4},
31         {5, 3, 3}
32     };
33
34     int result = findMaxRepeated(arr, 3, 3);
35
36     if (result != -1) {
37         cout << "Максимальний з повторюваних елементів: " << result << endl;
38     } else {
39         cout << "Немає повторюваних елементів" << endl;
40     }
41
42     return 0;
43 }

```

Завдання 3:

```

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> r(N);
12     for (int i = 0; i < N; ++i) {
13         cin >> r[i];
14     }
15
16     int a, b, c;
17     cin >> a >> b >> c;
18
19     r.erase(remove(r.begin(), r.end(), a), r.end());
20     r.erase(remove(r.begin(), r.end(), b), r.end());
21     r.erase(remove(r.begin(), r.end(), c), r.end());
22
23     int N_new = r.size();
24
25     if (N_new < 2) {
26         cout << 0 << endl;
27         return 0;
28     }
29
30     vector<int> sum_array(N_new - 1);
31     for (int i = 0; i < N_new - 1; ++i) {
32         sum_array[i] = r[i] + r[i + 1];
33     }
34
35     cout << N_new - 1 << endl;
36     for (int i = 0; i < N_new - 1; ++i) {
37         cout << sum_array[i] << " ";
38     }
39
40     return 0;
41 }

```

Завдання 4:

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int N, M;
8
9      cin >> N;
10     vector<int> a(N);
11     for (int i = 0; i < N; ++i) {
12         cin >> a[i];
13     }
14
15     cin >> M;
16     vector<int> b(M);
17     for (int i = 0; i < M; ++i) {
18         cin >> b[i];
19     }
20
21     bool in_a[101] = {false};
22     bool in_b[101] = {false};
23
24     for (int i = 0; i < N; ++i) {
25         in_a[a[i]] = true;
26     }
27
28     for (int i = 0; i < M; ++i) {
29         in_b[b[i]] = true;
30     }
31
32     int common_count = 0;
33     int unique_count = 0;
34
35     for (int i = 0; i <= 100; ++i) {
36         if (in_a[i] && in_b[i]) {
37             common_count++;
38         }
39         if (in_a[i] || in_b[i]) {
40             unique_count++;
41         }
42     }
43
44     cout << common_count << endl;
45     cout << unique_count << endl;
46
47     return 0;
48 }

```

Завдання 5

```

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     int original = number, reversed = 0;
21     while (number > 0) {
22         int digit = number % 10;
23         reversed = reversed * 10 + digit;
24         number /= 10;
25     }
26     return original == reversed;
27 }
28
29 int main() {
30     string str;
31
32     cout << "Введіть рядок: ";
33     cin >> str;
34     if (isPalindrome(str, 0, str.size() - 1)) {
35         cout << "Рядок " << str << " є паліндромом" << endl;
36     } else {
37         cout << "Рядок " << str << " не є паліндромом" << endl;
38     }
39
40     int number;
41
42     cout << "Введіть число: ";
43     cin >> number;
44     if (isPalindrome(number)) {
45         cout << "Число " << number << " є паліндромом" << endl;
46     } else {
47         cout << "Число " << number << " не є паліндромом" << endl;
48     }
49
50     return 0;
51 }

```

Завдання 6

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main() {
8      int k, n;
9
10     cin >> k >> n;
11
12     vector<int> prices(n);
13     for (int i = 0; i < n; i++) {
14         cin >> prices[i];
15     }
16
17     sort(prices.begin(), prices.end());
18
19     int count = 0;
20
21     for (int price : prices) {
22         if (k >= price) {
23             k -= price;
24             count++;
25         } else break;
26     }
27     cout << count;
28
29     return 0;
30 }

```

4) Результат виконання завдань, тестування та фактично затрачений час:

Завдання 1:

```

Введіть довжину масиву: 5
Введіть елементи масиву: 1 2 3 4 5
Введіть номер K (від 0 до 4): 2
Масив вліво від K до K-1: 3 4 5 1 2
Масив вправо від K до K+1: 2 1 1 5 4 3 5 4 3

```

Фактично затрачений час: 1 год 20 хв.

Завдання 2:

Максимальний з повторюваних елементів: 3

[Done] exited with code=0 in 0.722 seconds

Фактично затрачений час: 60 хв.

Завдання 3:

```
5
5 4 3 2 1
4 5 6
2
5 3 %
```

Фактично затрачений час: 1 год. 15 хв.

Завдання 4:

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

Фактично затрачений час: 1 год. 10 хв.

Завдання 5:

```
Введіть рядок для перевірки: radar
Рядок radar є паліндромом
Введіть число для перевірки: 123321
Число 123321 є паліндромом
```

```
Введіть рядок для перевірки: back
Рядок back не є паліндромом
Введіть число для перевірки: 453
Число 453 не є паліндромом
```

Фактично затрачений час: 1 год. 30 хв.

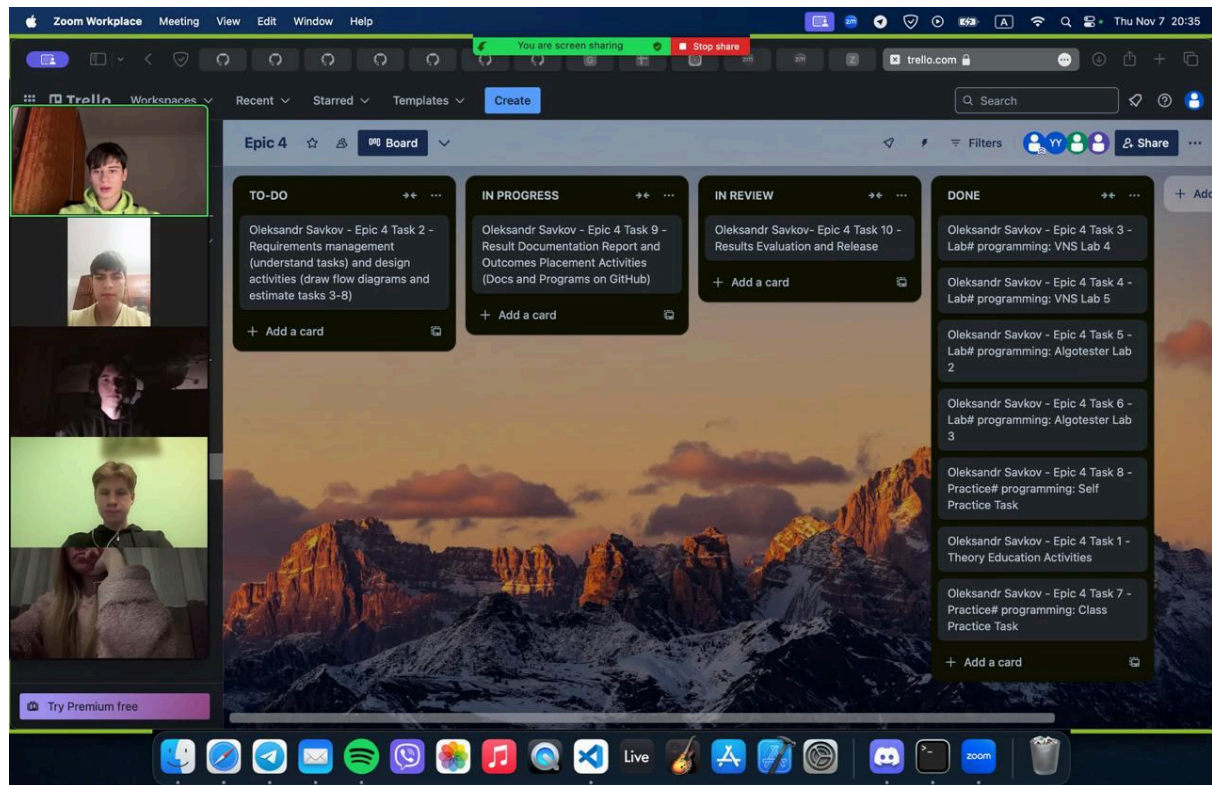
Завдання 6:

C++ 23	Зараховано	0.003	1.039
--------	------------	-------	-------

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40265#mySolutions>

Фактично затрачений час: 45 хв.

Робота з командою:



Висновок:

Під час виконання цього епіку я вивчив роботу з даними в C++, зокрема одновимірні та двовимірні масиви, а також їх використання для зберігання та обробки даних. Я засвоїв, як потрібно працювати з вказівниками та посиланнями, створювати та використовувати динамічні масиви. Окрім цього, я ознайомився зі створенням структур даних та вкладених структур, а також освоїв алгоритми для ефективної обробки масивів та структур, що дозволило писати більш складні та гнучкі програми.