

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 ШІ-13  
Бойко Роман Андрійович

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:** Розібратися з різними видами масивів, також навчитися використовувати вказівники та зрозуміти що таке структури і які вони бувають.

### Теоретичні відомості:

- масиви
- динамічні масиви
- структури
- вказівники

Джерела:

- [С++ • Теорія • Урок 40 • Одновимірні масиви](#)
- [С++ • Теорія • Урок 41 • Багатовимірні масиви](#)
- [С++ • Теорія • Урок 42 • Масиви рядки](#)
- [С++ • Теорія • Урок 57 • Вказівники. Частина 1](#)
- [С++ Теорія • Урок 63 • struct](#)

## Виконання роботи

**Особистий варіант - VNS - 1, Algotester Lab 2 - 3, Algotester Lab 3 - 2**

### Завдання 1: Practice task

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:

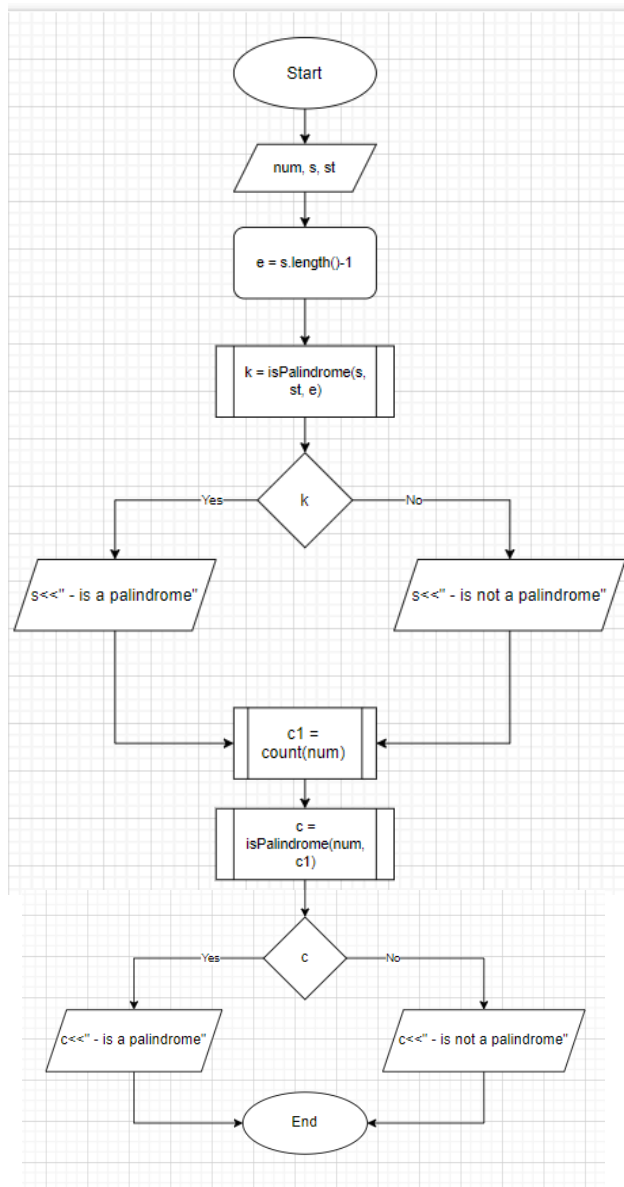
1. Визначення функції:
  - a. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:

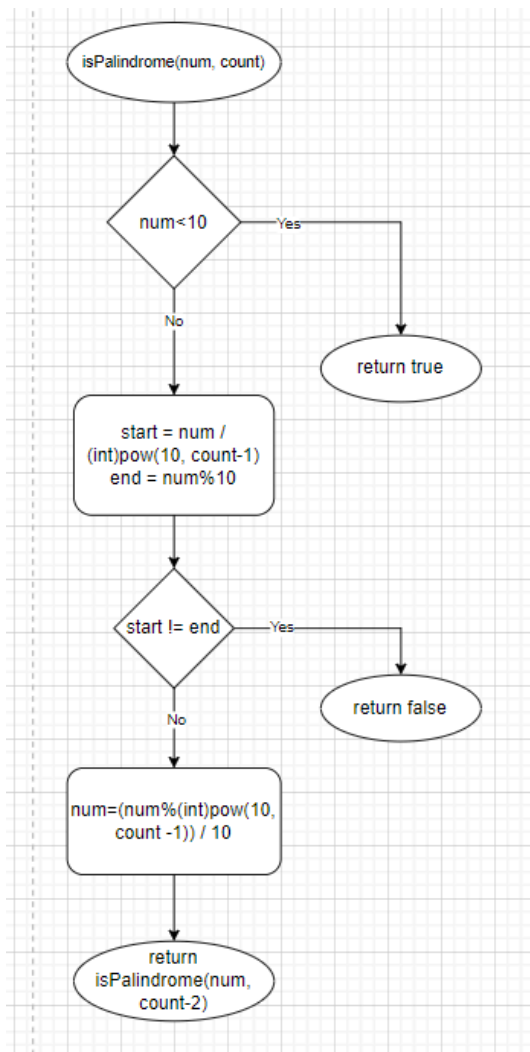
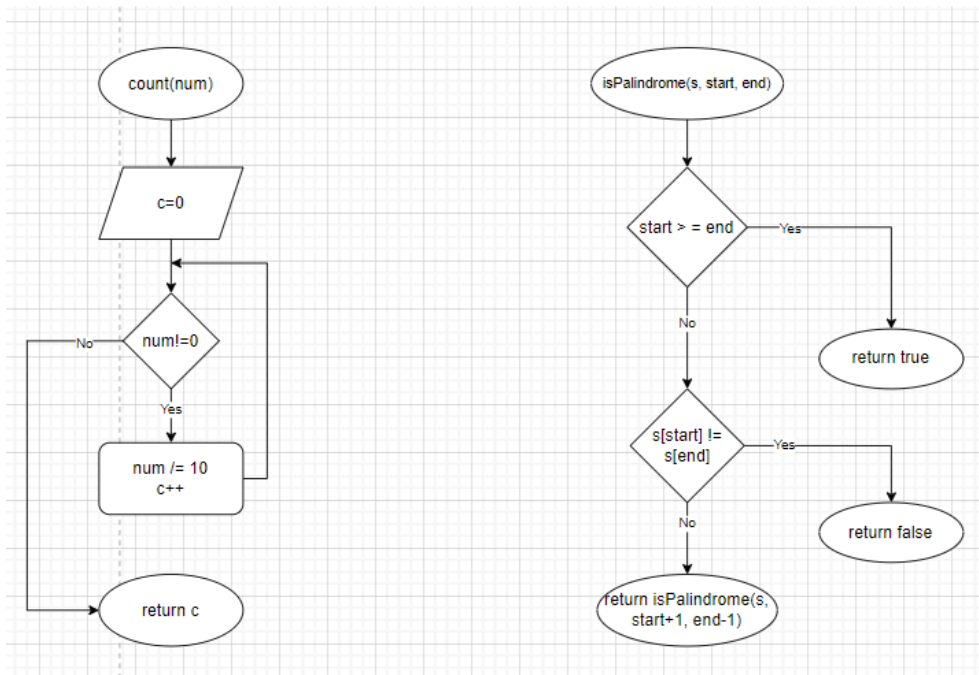
- a. `bool isPalindrome(const string& str, int start, int end);`
3. Перевантаження функцій:
  - a. Перевантажте функцію `isPalindrome` для роботи з цілими значеннями.
  - b. `bool isPalindrome(ціле число);`
4. Рекурсія:
  - a. Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій 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.

## Блок-схема до задачі:





## Розв'язок задачі:

```
1  #include <iostream>
2  #include <string>
3  #include <cmath>
4  using namespace std;
5  bool isPalindrome(string s, int start, int end);
6  bool isPalindrome(int num, int count);
7  int count(int num);
8  int main()
9  {
10     int num = 12329;
11     string s = "radar";
12     int st = 0;
13     int e = s.length() - 1;
14     bool k = isPalindrome(s, st, e);
15     if (k)
16     {
17         cout << s << " - is a palindrome!";
18     }
19     else
20     {
21         cout << s << " - is not a palindrome!";
22     }
23     cout << endl;
24     int c1 = count(num);
25     bool c = isPalindrome(num, c1);
```

```
26     if (c)
27     {
28         cout << num << " - is a palindrome!";
29     }
30     else
31     {
32         cout << num << " - is not a palindrome!";
33     }
34 }
35 int count(int num)
36 {
37     int c = 0;
38     while (num != 0)
39     {
40         num /= 10;
41         c++;
42     }
43     return c;
44 }
45 bool isPalindrome(string s, int start, int end)
46 {
```

```

47     if (start >= end)
48     {
49         return true;
50     }
51     if (s[start] != s[end])
52     {
53         return false;
54     }
55     return isPalindrome(s, start + 1, end - 1);
56 }
57 bool isPalindrome(int num, int count)
58 {
59     if (num < 10)
60     {
61         return true;
62     }
63     int start = num / (int)pow(10, count - 1);
64     int end = num % 10;
65     if (start != end)
66     {
67         return false;
68     }
69     num = (num % (int)pow(10, count - 1)) / 10;
70     return isPalindrome(num, count - 2);
71 }

```

### Вивід в терміналі:

```

radar - is a palindrome!
12329 - is not a palindrome!

```

**Час виконання завдання: ~ 60 хвилин**

## Завдання 2: VNS Lab 4

- 1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.
- 2) Роздрукувати отриманий масив.
- 3) Знищити елемент із номером K.
- 4) Додати після кожного парного елемента масиву елемент зі значенням 0.
- 5) Роздрукувати отриманий масив.

### Розв'язок задачі:

```

1  #include <iostream>
2  #include <random>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main()
8  {
9      srand(time(0));
10     int n;
11     cout << "Enter size of array: ";
12     cin >> n;
13     int arr[n];
14     for (int i = 0; i < n; i++)
15     {
16         arr[i] = rand() % 101;
17     }
18     for (int i = 0; i < n; i++)
19     {
20         cout << arr[i] << " ";
21     }
22     int k;
23     cout << "\nEnter index of element to remove: ";
24     cin >> k;

```

```

25     if (k >= 0 && k <= n)
26     {
27         for (int i = k; i < n - 1; i++)
28         {
29             arr[i] = arr[i + 1];
30         }
31         n--;
32     }
33     for (int i = 0; i < n; i++)
34     {
35         cout << arr[i] << " ";
36     }
37     cout << endl;
38     int newArr[n * 2];
39     int l = 0;
40     for (int i = 0; i < n; i++)
41     {
42         newArr[l] = arr[i];
43         l += 1;
44         if (arr[i] % 2 == 0)
45         {
46             newArr[l++] = 0;
47         }
48     }
49     cout << "New array is\n";
50     for (int i = 0; i < l; i++)
51     {
52         cout << newArr[i] << " ";
53     }
54 }

```

### Вивід в терміналі:

```

Enter size of array: 8
100 46 24 30 39 32 35 16
Enter index of element to remove: 3
100 46 24 39 32 35 16
New array is
100 0 46 0 24 0 39 32 0 35 16 0

```

**Час виконання завдання: ~ 40 хвилин**

### Завдання 3: VNS Lab 5

У двовимірному масиві записані слова, що представляють собою послідовність цифр, що завершуються 0. Необхідно роздрукувати слова через кому, взявши надрукований рядок у дужки. Довжина друкованого рядка 60 символів. Добування слова оформити у вигляді функції.

Наприклад:

вихідні дані - 123023402303450

234450234567010

234455677670450

результат - (123,234,23,345)(23445,234567,1)(23445567767,45)

### Розв'язок задачі:

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  const int maxCol = 60;
5  const int maxRow = 3;
6  void words(char arr[maxCol]);
7  int main()
8  {
9      char arr[maxRow][maxCol] = {
10         "123023402303450",
11         "234450234567010",
12         "234455677670450"};
13
14     for (int j = 0; j < maxRow; j++)
15     {
16         words(arr[j]);
17     }
18 }
19 void words(char arr[maxCol])
20 {

```

```

21     string s = "";
22     for (int i = 0; i < maxCol && arr[i] != '\0'; i++)
23     {
24         if (arr[i] == '0')
25         {
26             s += ',';
27         }
28         else
29         {
30             s += arr[i];
31         }
32     }
33
34     if (!s.empty())
35     {
36         s.pop_back();
37     }
38
39     cout << "(" << s << ")";
40 }

```

**Вивід в терміналі:**

```
(123,234,23,345)(23445,234567,1)(23445567767,45)
```

**Час виконання завдання: ~ 30 хвилин**

## Завдання 4: Algotester Lab 2

Вам дано масив цілих чисел розміром  $N$ , на першій та останній клітинці розміщено по дрону. Вони одночасно взлітають. На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться. Тобто лівий дрон у першу секунду з клітинки з індексом 1 перелетить у клітинку з індексом  $a_1$ , тобто його наступна позиція



рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якщо 2 дрони опиняються в одній клітинці - ви виводите Collision.

Якщо лівий дрон опиниться справа від правого - це Miss

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках  $a_i$  та  $a_{i+1}$  - виведіть

Stopped

Врахуйте, що перевіряти треба також до взльоту.

### Вхідні дані

У першому рядку ціле число  $N$  - розмір масиву

У другому рядку  $N$  цілих чисел - елементи масиву

### Вихідні дані

У першому рядку фінальна позиція першого та другого дрона.

У другому рядку одне зі слів:

Collision

Miss

Stopped

### Обмеження

$$1 \leq N \leq 1000$$

$$1 \leq a_i \leq 5$$

Гарантується, що дрони не можуть вилетіти за межі  $[1, N]$  (тобто не існує масиву, в якому дрони вилітають за межі до того, як станеться одна з 3-х вищезазначених ситуацій) та їх швидкість завжди більша за нуль.

### Розв'язок задачі:

```
1  #include <iostream>
2  using namespace std;
3  void Drones(int n, int arr[]);
4  int main()
5  {
6      int N;
7      cin >> N;
8      int arr[N];
9      for (int i = 1; i < N + 1; i++)
10     {
11         cin >> arr[i];
12     }
13     Drones(N, arr);
14 }
15 void Drones(int n, int arr[])
16 {
17     int leftDrone = 1;
18     int rightDrone = n;
19     while (true)
20     {
21         if (leftDrone == rightDrone)
22         {
23             cout << leftDrone << " " << rightDrone << endl;
24             cout << "Collision" << endl;
25             return;
26         }
27     }
28 }
```

```

27         else if (leftDrone + 1 == rightDrone)
28         {
29             cout << leftDrone << " " << rightDrone << endl;
30             cout << "Stopped" << endl;
31             return;
32         }
33         else if (leftDrone > rightDrone)
34         {
35             cout << leftDrone << " " << rightDrone << endl;
36             cout << "Miss" << endl;
37             return;
38         }
39         leftDrone += arr[leftDrone];
40         rightDrone -= arr[rightDrone];
41     }
42 }

```

**Вивід в терміналі:**

```

10
1 3 1 1 5 1 1 2 1 2
5 6
Stopped

```

**Час виконання завдання: ~ 30 хвилин**

## Завдання 5: Algotester Lab 3

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.

### Вхідні дані

У першому рядку ціле число N

у другому рядку N цілих чисел  $a_1..a_n$

У третьому рядку ціле число M

у четвертому рядку M цілих чисел  $b_1..b_n$

### Вихідні дані

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).

### Обмеження

$$0 \leq N \leq 100$$

$$0 \leq a_i \leq 100$$

$$0 \leq M \leq 100$$

$$0 \leq b_i \leq 100$$

## Розв'язок задачі:

```
1  #include <iostream>
2  #include <algorithm>
3  using namespace std;
4
5  void duplicates(int arr[], int nums, int dubl)
6  {
7      for (int i = 0; i < nums; i++)
8      {
9          for (int j = i + 1; j < nums; j++)
10         {
11             if (arr[i] == arr[j])
12             {
13                 cout << arr[i] << " - " << arr[j] << endl;
14                 dubl++;
15                 for (int k = j; k < nums - 1; k++)
16                 {
17                     arr[k] = arr[k + 1];
18                 }
19                 break;
20             }
21             nums--;
22             j--;
23         }
24     }
25 }
26
27 int main()
28 {
29     int N, M, k = 0, l = 0, f = 0;
30     cin >> N;
31     int arr1[N];
32     for (int i = 0; i < N; i++)
33     {
34         cin >> arr1[i];
35     }
36     cin >> M;
```

```
37     int arr2[M];
38     for (int i = 0; i < M; i++)
39     {
40         cin >> arr2[i];
41     }
42
43     int N1 = N;
44     duplicates(arr1, N1, f);
45
46     int M1 = M;
47     duplicates(arr2, M1, f);
48
49     for (int i = 0; i < N; i++)
50     {
51         for (int j = 0; j < M; j++)
52         {
53             if (arr1[i] == arr2[j])
54             {
55                 k += 1;
56             }
57         }
58     }
59
60     cout << endl;
61     cout << k << endl;
62     cout << N1 + M1 - k;
63 }
```

## Вивід в терміналі:

```
5
1 2 3 4 5
6
2 1 2 4 5 7

5
6
```

**Час виконання завдання: ~ 40 хвилин**

## Завдання 6: Algotester Lab 2 Variant 2

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.

## Розв'язок задачі:

```
1  #include <iostream>
2  #include <algorithm>
3  using namespace std;
4
5  void deleteElem(int arr[], int &n, int a, int b, int c);
6  void makeSum(int arr[], int newArr[], int n);
7
8  int main()
9  {
10     int N, a, b, c;
11     cin >> N;
12     int arr[N];
13
14     int *ptr = arr;
15     for (int i = 0; i < N; i++)
16     {
17         cin >> arr[i];
18     }
19     cin >> a >> b >> c;
20     deleteElem(arr, N, a, b, c);
21     if (N <= 1)
22     {
23         cout << 0;
24     }
25     else
26     {
27         int newArr[N];
28         makeSum(arr, newArr, N);
```

```

29         cout << N - 1 << endl;
30         for (int i = 0; i < N - 1; i++)
31         {
32             cout << newArr[i] << " ";
33         }
34     }
35 }
36
37 void deleteElem(int arr[], int &n, int a, int b, int c)
38 {
39     int newn = 0;
40     for (int i = 0; i < n; i++)
41     {
42         if (arr[i] != a && arr[i] != b && arr[i] != c)
43         {
44             arr[newn++] = arr[i];
45         }
46     }
47     n = newn;
48 }
49
50 void makeSum(int arr[], int newArr[], int n)
51 {
52     for (int i = 0; i < n - 1; i++)
53     {
54         newArr[i] = arr[i] + arr[i + 1];
55     }
56 }

```

**Вивід в терміналі:**

```

6
1 2 3 4 5 7
4 5 6
3
3 5 10

```

**Час виконання завдання: ~ 60 хвилин**

## Завдання 7: Algotester Lab 3 Variant 3

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 Scompressed

### Розв'язок задачі:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      string s, s1;
7      int l = 0;
8      cin >> s;
9      int k = 1;
10     for (int i = 1; i < s.size(); i++)
11     {
12         if (s[i] == s[i - 1])
13         {
14             k++;
15         }
16         else
17         {
18             s1 += s[i - 1];
19             if (k > 1)
20             {
21                 s1 += to_string(k);
22             }
23             k = 1;
24         }
25     }
26     s1 += s[s.length() - 1];
27     if (k > 1)
28     {
29         s1 += to_string(k);
30     }
31     cout << s1;
32 }
```

### Вивід в терміналі:

```
fffjgkhkkk11sdfpj
f3jgkhk312sdfpj
```

Час виконання завдання: ~ 60 хвилин

## Завдання 8: Self Task

Task: Student Management System

1. Define a `Student` structure that contains:
  - `string name`: The name of the student.
  - `int age`: The age of the student.
  - `float grade`: The average grade of the student.
  - `int id`: A unique ID for each student.
2. Implement a dynamic array of `Student` structures to store a list of students. Use pointers and dynamic memory allocation.
3. Write functions to:
  - Add a student to the dynamic array. (Resize the array if needed.)
  - Display all students in the array, including their names, ages, grades, and IDs.
  - Calculate the average grade of all students in the array.
4. Main Program:
  - In the main function, create a menu to allow the user to:
    - Add a new student.
    - Display the list of students.
    - Show the average grade of all students.
    - Exit the program.

## Розв'язок задачі:

```
1  #include <iostream>
2  using namespace std;
3
4  struct Student
5  {
6      string name;
7      int age;
8      int id;
9      float grade;
10 };
11
12 void AddStudents(Student *st, int &size);
13 void ShowStudents(Student *st, int size);
14 float ShowAverageGrade(Student *st, int size);
15 int main()
16 {
17     int size = 0;
18     Student *students = nullptr;
19
20     int ch;
21     while (true)
22     {
23         cout << "1 - Add a new student\n2 - Display the list of students\n3 - Show the average grade of all students\n4 - Exit the program.";
24         cin >> ch;
25         if (ch == 1)
26         {
27             AddStudents(students, size);
28         }
29         else if (ch == 2)
```

```

30     {
31         ShowStudents(students, size);
32     }
33     else if (ch == 3)
34     {
35         cout << "Average grade of all students: " << ShowAverageGrade(students, size) << endl;
36     }
37     else if (ch == 4)
38     {
39         break;
40     }
41 }
42 delete[] students;
43 }
44
45 void AddStudents(Student *&st, int &size)
46 {
47     size += 1;
48     Student *arr = new Student[size];
49     for (int i = 0; i < size - 1; i++)
50     {
51         arr[i] = st[i];
52     }
53     cout << "Enter student's name: ";
54     cin >> arr[size - 1].name;

```

```

54     cin >> arr[size - 1].name;
55     cout << "Enter student's age: ";
56     cin >> arr[size - 1].age;
57     cout << "Enter student's grade: ";
58     cin >> arr[size - 1].grade;
59     cout << "Enter student's id: ";
60     cin >> arr[size - 1].id;
61     delete[] st;
62     st = arr;
63 }
64
65 void ShowStudents(Student *st, int size)
66 {
67     if (size == 0)
68     {
69         cout << "We don't have any student!" << endl;
70         return;
71     }
72     for (int i = 0; i < size; i++)
73     {
74         cout << "Student " << i + 1 << endl;
75         cout << "Name: " << st[i].name << endl;
76         cout << "Age: " << st[i].age << endl;
77         cout << "Grade: " << st[i].grade << endl;
78         cout << "ID: " << st[i].id << endl;
79     }
80 }

```



```

82 float ShowAverageGrade(Student *st, int size)
83 {
84     float sum = 0;
85     if (size == 0)
86     {
87         return 0.0;
88     }
89     for (int i = 0; i < size; i++)
90     {
91         sum += st[i].grade;
92     }
93     return sum / size;
94 }
95

```

## Вивід в терміналі:

```

1 - Add a new student
2 - Display the list of students
3 - Show the average grade of all students
4 - Exit the program.1
Enter student's name: Ivan
Enter student's age: 12
Enter student's grade: 23
Enter student's id: 123213
1 - Add a new student
2 - Display the list of students
3 - Show the average grade of all students
4 - Exit the program.2
Student 1
Name: Ivan
Age: 12
Grade: 23
ID: 123213
1 - Add a new student
2 - Display the list of students
3 - Show the average grade of all students
4 - Exit the program.3
Average grade of all students: 23
1 - Add a new student
2 - Display the list of students
3 - Show the average grade of all students
4 - Exit the program.1
Enter student's name: Roman
Enter student's age: 18
Enter student's grade: 76
Enter student's id: 1124214

```

```

1 - Add a new student
2 - Display the list of students
3 - Show the average grade of all students
4 - Exit the program.3
Average grade of all students: 49.5
1 - Add a new student
2 - Display the list of students
3 - Show the average grade of all students
4 - Exit the program.2
Student 1
Name: Ivan
Age: 12
Grade: 23
ID: 123213
Student 2
Name: Roman
Age: 18
Grade: 76
ID: 1124214
1 - Add a new student
2 - Display the list of students
3 - Show the average grade of all students
4 - Exit the program.4

```

## Висновок:

У цьому епіку я навчився працювати зі структурами, розібрав що таке вказівники та динамічні масиви. Також попрацював із масивами різних видів та стрічками.