**Лабараторная работа № 6**

*«Написание программ с использованием одномерных массивов и алгоритмов их обработки»*

**Вариант 8**

***Цель работы:***освоить методы разработки алгоритмов и программ решения задач, предназначенных для формирования и обработки одномерных  массивов.

**Задание 1.** Создать три массива из 5 элементов каждый:

1. Первый массив инициализировать при объявлении
2. Второй массив заполнить с клавиатуры
3. Третий массив заполнить случайными числами

Создать четвертый массив, элементы которого будут равны произведению соответствующих элементов первых трех массивов.

Вывести все массивы.

**Решение:**

void task1() {

const size\_t SIZE = 5;

int arr1[SIZE] = {1, 2, 3, 4, 5}, arr2[SIZE], arr3[SIZE], arr4[SIZE];

for (size\_t k = 0; k < SIZE; ++k) {

printf("Enter %ld element of array: ", k + 1);

std::cin >> arr2[k];

arr3[k] = rand() % 1000;

arr4[k] = arr1[k] \* arr2[k] \* arr3[k];

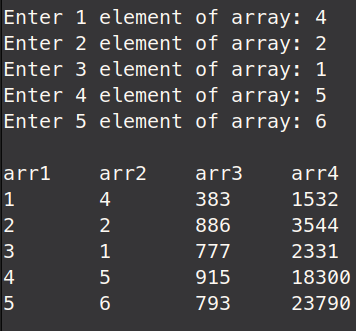
}

printf("\narr1\tarr2\tarr3\tarr4\n");

for (size\_t k = 0; k < SIZE; ++k)

printf("%d\t%d\t%d\t%d\n", arr1[k], arr2[k], arr3[k], arr4[k]);

}

**Задание 2.** Создать массив из 20 элементов. Заполнить его случайными числами. Определить, сколько элементов массива больше двух своих соседей. Вывести позиции таких элементов и их общее количество.

**Решение:**

void task2() {

const size\_t SIZE = 20;

int arr[SIZE];

for (size\_t k = 0; k < SIZE; ++k)

arr[k] = rand() % 1000;

size\_t counter = 0;

for (size\_t k = 1; k < SIZE - 1; ++k) {

if (arr[k] > arr[k - 1] && arr[k] > arr[k + 1]) {

std::cout << k << ' ';

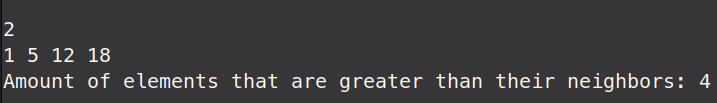
++counter;

}

}

std::cout << "\nAmount of elements that are greater than their neighbors: " << counter << '\n';

}



**Задание 3.** Создать и инициализировать массив символов char большими английскими буквами (массив обязательно должен содержать вашу фамилию). Размер массива от 7 символов до скольки хотите. Зашифровать и расшифровать массив шифром Цезаря с цикл. сдвигом 4.

**Решение:**

void task3() {

const char surname[] = {"KLESHCHINSKY"};

const size\_t size = sizeof(surname) / sizeof(surname[0]);

std::cout << "Original: " << surname << '\n';

char code[size], decode[size];

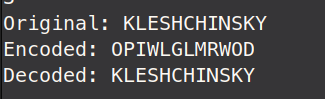
std::cout << "Encoded: ";

for (size\_t k = 0; k < size; ++k) {

char encoded = surname[k] + 4;

code[k] = (encoded > 'Z' ? 'A' + (encoded - 'Z') : encoded);

std::cout << code[k];

}

std::cout << std::endl;

std::cout << "Decoded: ";

for (size\_t k = 0; k < size; ++k) {

char decoded = code[k] - 4;

decode[k] = (decoded < 'A' ? 'Z' - ('A' - decoded) : decoded);

std::cout << decode[k];

}

std::cout << std::endl;

}