

Спецификация лабораторной работы № 3

Данная программа создает массивы с целыми числами и применяет к ним функцию, которая преобразовывает массив, располагая его значения в особом порядке, указанном в варианте. С начала располагаются четные, а потом нечетные элементы.

5. Массив целых чисел преобразовать таким образом, чтобы с начала располагались четные элементы, а потом нечетные

Тестовый набор с выходными данными

В input.txt предложен вариант массивов для решения 3 пункта.

```
Объявленный массив
1 2 10 5 21
Решение
2 10 1 5 21
Динамический массив чисел
Введите количество элементов в динамическом массиве:
2
8 4
Решение
8 4
1
2
3
4
5
Решение массивов из файла
2 4 1 3 5
4
8
7
10
11
12
2
3
31
19
Решение массивов из файла
4 8 10 12 2 7 11 3 31 19
5
25
20
40
1
3
53
Решение массивов из файла
20 40 5 25 1 3 53
```

Объявленный массив

1 2 10 5 21

Решение

2 10 1 5 21

Динамический массив чисел

Введите количество элементов в динамическом массиве:

5

8 7 0 3 3

Решение

8 0 7 3 3

1

2

3

4

5

Решение массивов из файла

2 4 1 3 5

4

8

7

10

11

12

2

3

31

19

Решение массивов из файла

4 8 10 12 2 7 11 3 31 19

5

25

20

40

1

3

53

Решение массивов из файла

20 40 5 25 1 3 53

```

Объявленный массив
1 2 10 5 21
Решение
2 10 1 5 21
Динамический массив чисел
Введите количество элементов в динамическом массиве:
3
5 0 4
Решение
0 4 5
1
2
3
4
5
Решение массивов из файла
2 4 1 3 5
4
8
7
10
11
12
2
3
31
19
Решение массивов из файла
4 8 10 12 2 7 11 3 31 19
5
25
20
40
1
3
53
Решение массивов из файла
20 40 5 25 1 3 53

```

Программа:

```

#include <iostream>
#include <ctime>
#pragma warning(disable : 4996) // Обход безопасности и доступ к файлу freopen

void solveFunction(int *array, int size);

int randomFunction(int* dynamicArray, int dynamicArraySize);

int main()
{
    setlocale(LC_ALL, "ru");
    srand(static_cast<unsigned int>(time(nullptr)));

    const int STATIC_SIZE = 5;
    int staticArray[STATIC_SIZE] = { 1, 2, 10, 5, 21 };
    std::cout << "Объявленный массив \n";

    for (int i = 0; i < STATIC_SIZE; i++)
    {
        std::cout << staticArray[i] << " ";
    }
}

```

```

    }
    std::cout << "\nРешение \n";
    solveFunction(staticArray, STATIC_SIZE);

    int dynamicArraySize;
    std::cout << "\nДинамический массив чисел";
    std::cout << "\nВведите количество элементов в динамическом массиве: " <<
std::endl;
    std::cin >> dynamicArraySize;
    if (std::cin.fail())
    {
        std::cout << "Вы не ввели количество элементов для динамического массива.";
    }
    int* dynamicArray = new int[dynamicArraySize];
    randomFunction(dynamicArray, dynamicArraySize);
    std::cout << "\nРешение \n";
    solveFunction(dynamicArray, dynamicArraySize);
    delete[]dynamicArray;

    freopen("input.txt", "r", stdin);
    int size;
    while (std::cin >> size)
    {
        int* array = new int[size];
        for (int i = 0; i < size; i++)
        {
            std::cin >> array[i];
            std::cout << "\n" << array[i] << " ";
        }
        std::cout << "\nРешение массивов из файла \n";
        solveFunction(array, size);
        delete[]array;
    }
    return 0;
}

void solveFunction(int *array, int size)
{
    for (int i = 0; i < size - 1; i++)
    {
        for (int j = 0; j < size - 1; j++)
        {
            if ((array[j] % 2 != 0) && (array[j + 1] % 2 == 0))
            {
                std::swap(array[j], array[j + 1]);
            }
        }
    }
    for (int i = 0; i < size; i++)
    {
        std::cout << array[i] << " ";
    }
}

int randomFunction(int* dynamicArray, int dynamicArraySize)
{
    for (int i = 0; i < dynamicArraySize; i++)
    {
        dynamicArray[i] = rand() % 10;
        std::cout << dynamicArray[i] << " ";
    }
    return 0;
}

```