



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт искусственного интеллекта  
Кафедра общей информатики

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**  
**по дисциплине**  
**«Структуры и алгоритмы обработки данных»**

Выполнил студент группы ИКБО-13-21

Дамарад Д.В.

Принял

старший преподаватель

Скворцова Л.А.

Практическая  
работа выполнена  
«Зачтено»

«\_\_»\_\_\_\_\_2021 г.

«\_\_»\_\_\_\_\_2021 г.

Москва 2022

## СОДЕРЖАНИЕ

<b>1</b>	<b><i>ПОСТАНОВКА ЗАДАЧИ</i></b> .....	<b>3</b>
<b>2</b>	<b><i>ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ</i></b> .....	<b>4</b>
<b>3</b>	<b><i>ВЫВОДЫ</i></b> .....	<b>17</b>

## **1 ПОСТАНОВКА ЗАДАЧИ**

Дан статический массив из целых беззнаковых элементов.

1. Вставить новый элемент в массив перед элементом, у которого четное количество цифр.
2. Удалить все четные числа массива.
3. Найти максимальное число среди элементов массива, расположенных на четных местах.

## 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

### Декомпозиция

Задачу следует разбить на следующие подзадачи:

- Ввод и вывод массива(вектора)
- Проверка четное или нечетное количество чисел.
- Вставка нового элемента в массив перед элементом, у которого четное количество цифр.
- Удаление всех четных чисел массива.
- Поиск максимального числа среди элементов массива,

расположенных на четных местах.

Определение функций (на примере статического массива):

1. Заполнять массив будем с помощью функции

Предусловие: arr - неинициализированный массив n, число элементов size

Постусловие: заполненный массив

void Fill\_array(int arr[], int size,int n)

2. Вывод массива на экран

Предусловие: массив arr, число элементов n

Постусловие: вывод значений массива

void Show\_array(int arr[], int size)

3. Проверка четное или нечетное количество чисел.

Предусловие: число a

Постусловие: целочисленная

переменная

int asd(int a)

4. Вставка нового элемента в массив перед элементом, у которого четное количество цифр.

Предусловие: массив целых чисел arr, длина size, новый элемент newelement.

Постусловие: измененный массив.

```
void count(int arr[], int size, int newelement)
```

5. Удаление всех четных чисел массива.

Предусловие: массив целых чисел arr, длина size.

Постусловие: измененный массив.

```
void delete_even(int arr[], int size)
```

6. Поиск максимального числа среди элементов массива, расположенных на четных местах.

Предусловие: массив целых чисел arr, длина size.

Постусловие: целочисленная переменная

```
void insert(int arr[], int size)
```

Алгоритмы задач декомпозиции:

**Алгоритм void Fill\_array(int arr[], int size,int n)**

```
void Fill_array(int arr[], int size,int n)
{
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
        if (arr[i] < 0)
        {
            cout << "Число не может быть меньше 0, для данного элемента будет
выбрано значение 0" << endl;
            arr[i] = 0;
        }
    }
}
```

**Алгоритм void Show\_array(int arr[], int size)**

```
void Show_array(int arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
}
```

**Алгоритм int asd(int a)**

```
int asd(int a){
int c, i = 0;
while (a!= 0) {
    i++;
    a = a / 10;
}
c = i % 2;
return c;
}
```

**Алгоритм void count(int**

**arr[], int size, int  
newelement)**

```
void count(int arr[], int size, int newelement) {
    int d = 0, c;
```

```

for (int i = 0; i < size; i++) {
    if (arr[i] != 0) {
        if (asd(arr[i]) == 0) {
            c = arr[i];
            arr[i] = newelement;
            for (int j = i + 1; j < size - 1; j++) {
                d = arr[j];
                arr[j] = c;
                c = d;
                i++;
            }
        }
    }
}
}

```

**Алгоритм delete\_even(int arr[], int size)**

```

int findMaxIndex(int* arr, int len) {
for (int i = 0; i < size; i++) {
    if (arr[i] % 2 == 0) {
        arr[i] = 0;
    }
}
}

```

**Алгоритм void maxoneven(int arr[], int size)**

```

void maxoneven(int arr[], int size) {
    int max = 0;
    for (int i = 0; i < size; i++) {
        if (i % 2 == 0) {
            if (arr[i] > max) {
                max = arr[i];
            }
        }
    }
    cout << "Максимальное число среди элементов массива, расположенных на
четных местах: " << max << endl;
}

```

## Тесты функций:

### 1)asd

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	103	1	1	+
2	1338	0	0	+

### 2)Fill\_array

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	Arr, size=3	Заполненный на 3 числа массив	Заполненный на 3 числа массив	+
2	Arr, size=4	Заполненный на 4 числа массив	Заполненный на 4 числа массив	+

### 3)Show\_array

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	Arr = {1,2} size =2	1 2	1 2	+



2	Arr = {1,2,1235} size =2	1 2 1235	1 2 1235	+
---	-----------------------------	----------	----------	---

#### 4)count

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	arr1(100,11,33,433,535) newelement=5	arr2(100,5,11,33,433,535)	arr2(100,5,11,33,433,535)	+
2	arr1(31,444,34,445,1) newelement=7	arr2(7,31,444,34,445,1)	arr2(7,31,444,34,445,1)	+

#### 5)delete\_even

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	array1(1,2,3,4,5)	array1(1,0,3,0,5)	array1(1,0,3,0,5)	+
2	array1(11,22,33,44,77)	array1(11,0,33,0,77)	array1(11,0,33,0,77)	+

#### 6)maxoneven

Номер теста	Исходные данные	Ожидаемый результат	Результат программы	Тест пройден/не пройден
1	array(1,7,4,1,7)	7	7	+
2	array(1,2,3,4,5)	5	5	+

Стоит отметить, что все функции для реализации на динамическом массиве и векторе идентичны, изменены только входные данные функции.

## Полный код программы

### static.cpp

```
#include <iostream>
#include <vector>
#include <cstring>
#include <clocale>
#include <conio.h>
using namespace std;
int asd(int a) { // Проверка четное или нечетное количество чисел
    int c, i = 0;
    while (a != 0) {
        i++;
        a = a / 10;
    }
    c = i % 2;
    return c;
}
void Fill_array(int arr[], int size, int n) //Заполнение массива
{
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
        if (arr[i] < 0)
        {
            cout << "Число не может быть меньше 0, для данного элемента будет выбрано значение 0" << endl;
            arr[i] = 0;
        }
    }
}
void Show_array(int arr[], int size) // Вывод всех элементов массива
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
}
void count(int arr[], const int size, int newelement) { // вставка нового элемента в массив перед элементом, у которого четное количество цифр
    int d = 0, c;
    for (int i = 0; i < size; i++) {
        if (arr[i] != 0) {
            if (asd(arr[i]) == 0) {
                c = arr[i];
                arr[i] = newelement;
                for (int j = i + 1; j < size - 1; j++) {
                    d = arr[j];
                    arr[j] = c;
                    c = d;
                    i++;
                }
            }
        }
    }
}
```

```

}
void delete_even(int arr[], int size){ //Удаление всех четных чисел массива.
    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 == 0) {
            arr[i] = 0;
        }
    }
}
}
void maxoneven(int arr[],int size){ // поиск максимального числа среди элементов массива,
расположенных на четных местах.
    int max=0;
    for (int i = 0; i < size; i++) {
        if (i % 2 == 0) {
            if (arr[i] > max) {
                max = arr[i];
            }
        }
    }
    cout << "Максимальное число среди элементов массива, расположенных на четных местах: "
<< max << endl;
}
int main()
{
    setlocale(LC_ALL, "rus");
    const int size = 5;
    int arr1[size] = { 0 }, n;
    cout << "Введите количество обрабатываемых элементов массива: ";
    cin >> n;
    if (n > 0 && n <= size) {
        cout << "Введите " << n << " чисел " << endl;
        Fill_array(arr1, size, n);
    }
    else {
        cout << "n должно быть больше нуля";
        exit(0);
    }
    cout << "Исходный массив:" << endl;
    Show_array(arr1, size);
    cout << endl;
    int y;
    cout << "Введите номер задачи: ";
    cin >> y;
    switch (y) {
    case(1):
    {
        const int newsize = size + 1;
        int arr2[newsize] = { 0 };
        for (int i = 0; i < size; i++) {
            arr2[i]=arr1[i];
        }
        int newelement;
        cout << "Введите элемент, который желаете вставить перед элементом, у которого
четное количество цифр: ";
        cin >> newelement;
        count(arr2, newsize, newelement);
        cout << "Новый массив: " << endl;
        Show_array(arr2, newsize);
        break;
    }
    case(2):
    {
        delete_even(arr1, size);
        cout << "Новый массив: " << endl;
        Show_array(arr1, size);
        break;
    }
    case(3):

```

```

    {
        maxoneeven(arr1, size);
        break;
    }
}
return 0;
}

```

## dynamic.cpp

```

#include <iostream>
#include <vector>
#include <cstring>
#include <clocale>
#include <conio.h>
using namespace std;
void Fill_array(int arr[], int size, int n) //Заполнение массива
{
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
        if (arr[i] < 0)
        {
            cout << "Число не может быть меньше 0, для данного элемента будет выбрано
значение 0" << endl;
            arr[i] = 0;
        }
    }
}
void Show_array(int arr[], int size) // Вывод всех элементов массива
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
}
int asd(int a) { // проверка четное или нечетное количество чисел
    int c, i = 0;
    while (a != 0) {
        i++;
        a = a / 10;
    }
    c = i % 2;
    return c;
}
void count(int arr[], int size, int newelement) { // вставка нового элемента в массив перед
элементом, у которого четное количество цифр
    int d = 0, c;
    for (int i = 0; i < size; i++) {
        if (arr[i] != 0) {
            if (asd(arr[i]) == 0) {
                c = arr[i];
                arr[i] = newelement;
                for (int j = i + 1; j < size - 1; j++) {
                    d = arr[j];
                    arr[j] = c;
                    c = d;
                    i++;
                }
            }
        }
    }
}
void delete_even(int arr[], int size) { //Удаление всех четных чисел массива.
    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 == 0) {

```

```

        arr[i] = 0;
    }
}
}
void maxoneven(int arr[], int size) { // поиск максимального числа среди элементов массива,
расположенных на четных местах.
    int max = 0;
    for (int i = 0; i < size; i++) {
        if (i % 2 == 0) {
            if (arr[i] > max) {
                max = arr[i];
            }
        }
    }
    cout << "Максимальное число среди элементов массива, расположенных на четных местах: "
<< max << endl;
}
int main()
{
    setlocale(LC_ALL, "rus");
    int size;
    cout << "Введите размер массива: ";
    cin >> size;
    if (size <= 0) {
        cout << "Размер должен быть больше нуля";
        exit(0);
    }
    int* arr1 = new int[size], n;
    for (int i = 0; i < size; i++) {
        arr1[i] = 0;
    }
    cout << "Введите количество обрабатываемых элементов массива: ";
    cin >> n;
    if (n > 0 && n <= size) {
        cout << "Введите " << n << " чисел " << endl;
        Fill_array(arr1, size, n);
    }
    else {
        cout << "n должно быть больше нуля и не больше размера массива";
        exit(0);
    }
    cout << "Исходный массив:" << endl;
    Show_array(arr1, size);
    cout << endl;
    int y;
    cout << "Введите номер задачи: ";
    cin >> y;
    switch (y) {
        case(1): // Вставить новый элемент в массив перед элементом, у которого четное
количество цифр.
        {
            int newsize = size + 1;
            int* arr2 = new int[newsize];
            for (int i = 0; i < newsize; i++) {
                arr2[i] = 0;
            }
            for (int i = 0; i < size; i++) {
                arr2[i] = arr1[i];
            }
            int newelement;
            cout << "Введите элемент, который желаете вставить перед элементом, у которого
четное количество цифр: ";
            cin >> newelement;
            count(arr2, newsize, newelement);
            cout << "Новый массив: " << endl;
            Show_array(arr2, newsize);
            break;

```

```

    }
    case(2): // Удалить все четные числа массива.
    {
        delete_even(arr1, size);
        cout << "Новый массив: " << endl;
        Show_array(arr1, size);
        break;
    }
    case(3): // Найти максимальное число среди элементов массива, расположенных на четных
местах.
    {
        maxoneven(arr1, size);
        break;
    }
}
return 0;
}

```

## vector.cpp

```

#include <vector>
#include <iostream>
#include <cstring>
#include <locale>
#include <conio.h>
using namespace std;
void Fill_array(vector<int>& vector, int size, int n)
{
    for (int i = 0; i < n; i++)
    {
        cin >> vector[i];
        if (vector[i] < 0)
        {
            cout << "Число не может быть меньше 0, для данного элемента будет выбрано
значение 0" << endl;
            vector[i] = 0;
        }
    }
}

void Show_array(vector<int>& vector, int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << vector[i] << " ";
    }
}

int asd(int a) {
    int c, i = 0;
    while (a != 0) {
        i++;
        a = a / 10;
    }
    c = i % 2;
    return c;
}

void count(vector<int> &vector, int size, int newelement) {
    int d = 0, c;
    for (int i = 0; i < size; i++) {
        if (vector[i] != 0) {
            if (asd(vector[i]) == 0) {
                c = vector[i];
                vector[i] = newelement;
                for (int j = i + 1; j < size 141; j++) {
                    d = vector[j];

```

```

        vector[j] = c;
        c = d;
        i++;
    }
}
}
}
}
void delete_even(vector<int>& vector, int size) {
    for (int i = 0; i < size; i++) {
        if (vector[i] % 2 == 0) {
            vector[i] = 0;
        }
    }
}
void maxoneven(vector<int>& vector, int size) {
    int max = 0;
    for (int i = 0; i < size; i++) {
        if (i % 2 == 0) {
            if (vector[i] > max) {
                max = vector[i];
            }
        }
    }
    cout << "Максимальное число среди элементов массива, расположенных на четных местах: "
<< max << endl;
}
int main()
{
    setlocale(LC_ALL, "rus");
    int size, n;
    cout << "Введите размер массива: ";
    cin >> size;
    if (size <= 0) {
        cout << "Размер должен быть больше нуля";
        exit(0);
    }
    vector<int> myVector(size);
    cout << "Введите количество обрабатываемых элементов массива: ";
    cin >> n;
    if (n > 0 && n <= size) {
        cout << "Введите " << n << " чисел " << endl;
        Fill_array(myVector, size, n);
    }
    else {
        cout << "n должно быть больше нуля";
        exit(0);
    }
    cout << "Исходный массив:" << endl;
    Show_array(myVector, size);
    cout << endl;
    int y;
    cout << "Введите номер задачи: ";
    cin >> y;
    switch (y) {
    case(1):
    {
        int newsize = size + 1;
        vector<int> myVector2(newsize);
        for (int i = 0; i < size; i++) {
            myVector2[i] = myVector[i];
        }
        int newelement;
        cout << "Введите элемент, который желаете вставить перед элементом, у которого
четное количество цифр: ";
        cin >> newelement;
        count(myVector2, newsize, newelement);

```

```

        cout << "Новый массив: " << endl;
        Show_array(myVector2, newsize);
        break;
    }
    case(2):
    {
        delete_even(myVector, size);
        cout << "Новый массив: " << endl;
        Show_array(myVector, size);
        break;
    }
    case(3):
    {
        maxoneven(myVector, size);
        break;
    }
    }
    return 0;
}

```



### **3 ВЫВОДЫ**

В ходе работы на языке C++ было разработано приложение, реализующее задачи в соответствии с персональным вариантом в 3 вариациях: на статическом массиве, на динамическом массиве и на векторе.