



МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Радиотехнических и телекоммуникационных систем (РТС)

Кафедра Радиоэлектронных систем и комплексов (РЭСК)

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**ПО ДИСЦИПЛИНЕ «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ТЕХНИЧЕСКИХ
РАСЧЕТОВ»**

на тему

Освоение работы с одномерными массивами и функциями на языке C++

Вариант №5

Студент группы РКБО-01-20

Шевцов Е.С.

Преподаватель кафедры РЭСК

Авдеев К.В.

Задание 1

1. Текст программы:

```
#include <iostream>
using namespace std;

// simple - проверка, является ли число a простым
bool simple(int a)
{
    for (int i = a - 1; i > 1; i--)
    {
        if (a % i == 0)
        {
            return false;
        }
    }
    return true;
}

// printArr - вывод массива на экран в приятном для глаза виде
void printArr(int *arr, int size)
{
    cout << "[";
    for (int i = 0; i < size; i++)
    {
        if (i != 0)
            cout << ", ";
        cout << arr[i];
    }
    cout << "]" << endl;
}

int main()
{
    int N = 0, i = 0;
    cout << "Введите N: ";
    cin >> N;
    int arr[N];

    // цикл до максимального значения int16, с целью проверки каждого числа на простоту
    // и занесение простого числа в массив
    for (int a = 2; a < INT16_MAX; a++)
    {
        if (simple(a))
        {
            arr[i] = a;
            i++;
            // если достигли конца массива - выходим
            if (i == N)
                break;
        }
    }
    // выводим массив на экран
    printArr(arr, N);
}
```

2. Результат работы с тестовыми данными:

```
egor@MacBook-Pro-Egor LR2 % g++ 1.cpp -o tmp; ./tmp; rm tmp
Введите N: 6
[2, 3, 5, 7, 11, 13]
```

Задание 2

Оно бы уместилось, если бы не куча комментариев $_ \backslash _ (\text{ツ}) _ / _$

1. Текст программы(необходимо собирать со стандартом выше с++17):

```
#include <iostream>
#include <map>
using namespace std;

// setArray - ввод массива пользователем
void setArray(int *arr, int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << "Введите элемент " << i + 1 << ": ";
        cin >> arr[i];
    }
}

// countEqualElements - возвращает число повторений самого частовстречающегося элемента массива
int countEqualElements(int *a, int size)
{
    // Создаём карту в которой ключ - элемент из массива, значение - кол-во повторений этого элемента в массиве
    map<int, int> counts;
    // Подсчёт кол-во повторений для каждого элемента
    for (int i = 0; i < size; i++)
    {
        // Если элемента нет в мапе, то он добавляется и его значение равно 1(0++ = 1)
        // Если элемент есть в мапе, то к его значению добавляется 1
        counts[a[i]]++;
    }

    int max = 0;
    /*
    Начиная с C++11 можно проходить(спец символ :)
    через списки без использования индексов, в тех случаях когда он не нужен.
    Использование [тип_данных имя_переменной_которой_назначается_текущее_вхождение : список_через_которой_проходим]).
    Спец. слово auto говорит компилятору автоматически определить тип ключа и значения в мапе,
    что позволяет "проходить" через любые карты с разными типами данных
    */
    for (const auto &[key, value] : counts)
    {
        if (value > max)
        {
            max = value;
        }
    }
    return max;
}

int main()
{
    int N = 0;
    cout << "Введите размер N: ";
    cin >> N;
    int a[N], b[N];

    cout << endl << "Заполнение первого массива" << endl;
    setArray(a, N);
    cout << endl << "Заполнение второго массива" << endl;
    setArray(b, N);

    int countA = countEqualElements(a, N), countB = countEqualElements(b, N);
    if (countA > countB)
    {
        cout << "В первом массиве больше повторяющихся элементов." << endl;
    }
    else if (countA < countB)
    {
        cout << "Во втором массиве больше повторяющихся элементов." << endl;
    }
    else
    {
        cout << "В обоих массивах одинаковое кол-во повторяющихся элементов" << endl;
    }
}
```

2. Результат работы с тестовыми данными:

```
egor@MacBook-Pro-Egor LR2 % g++ --std=c++17 2.cpp -o tmp; ./tmp; rm tmp
Введите размер N: 6

Заполнение первого массива
Введите элемент 1: 1
Введите элемент 2: 1925
Введите элемент 3: 1925
Введите элемент 4: 3
Введите элемент 5: 1
Введите элемент 6: 1

Заполнение второго массива
Введите элемент 1: 1
Введите элемент 2: 4
Введите элемент 3: 5
Введите элемент 4: 6
Введите элемент 5: 44
Введите элемент 6: 44
В первом массиве больше повторяющихся элементов.
```

Задание 3

1. Текст программы:

```
#include <iostream>
#include <cmath>
using namespace std;

// moveCoord – вычисление новых координат X и Z
void moveCoord(float *X, float *Z, float fi)
{
    *X = (*X) * cos(fi) + (*Z) * sin(fi);
    *Z = (-1) * (*X) * sin(fi) + (*Z) * cos(fi);
}

// moveCoord – вычисление координаты Y
void moveCoord(float *Y, float A)
{
    *Y = (*Y) - A;
}

// moveCoord – вычисление новых координат после сдвига
void moveCoord(float *X, float *Y, float *Z, float fi, float A)
{
    moveCoord(X, Z, fi);
    moveCoord(Y, A);
}

// fillFloat – заполнение пользователем значение переменной с плавающей точкой
void fillFloat(float *a, string comment)
{
    cout << "Введите " << comment << ": ";
    cin >> *a;
}

int main()
{
    float X, Y, Z, fi, A;
    fillFloat(&X, "X");
    fillFloat(&Y, "Y");
    fillFloat(&Z, "Z");
    fillFloat(&fi, "угол φ");
    fillFloat(&A, "A");

    // преобразование градусов угла в радианы
    fi = fi * M_PI / 180.0;

    moveCoord(&X, &Y, &Z, fi, A);
    cout << "Новые координаты: (" << X << ", " << Y << ", " << Z << ")" << endl;
}
```

2. Результат работы с тестовыми данными:

```
egor@MacBook-Pro-Egor LR2 % g++ 3.cpp -o tmp; ./tmp; rm tmp
Введите X: 3.2
Введите Y: 1.5
Введите Z: 3
Введите угол φ: 90
Введите A: 2
Новые координаты: (3,-0.5,-3)
```