

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Департамент прикладной математики**

**ОТЧЕТ**  
**К ЛАБОРАТОРНОЙ РАБОТЕ 2**  
**по дисциплине «Алгоритмизация и программирование»**

Работу выполнила

студентка группы БПМ 173

\_\_\_\_\_

дата, подпись

М.В. Самоделкина

Работу проверил

\_\_\_\_\_

дата, подпись

С.А. Булгаков

Москва 2019

## Содержание

<b>Постановка задачи</b>	<b>3</b>
<b>1 Основная часть</b>	<b>4</b>
1.1 Общая идея решения задачи . . . . .	4
1.2 Структура и принципы действия . . . . .	4
1.2.1 Задание 1 . . . . .	4
1.2.2 Задание 2 . . . . .	4
1.2.3 Задание 3 . . . . .	5
1.3 Процедура получения исполняемых программных модулей . . . . .	5
1.4 Результаты тестирования . . . . .	5
<b>Приложение А</b>	<b>6</b>
Task5.cpp . . . . .	6

## Постановка задачи

Используя алгоритмы стандартной библиотеки *<algorithm>* и, при необходимости, *<functional>*, выполнить задания в соответствии с вариантом 18:

1. Ввести произвольную строку с клавиатуры и удалить все пробелы. Ввести слово и, если оно не встречается в ранее введенной строке, вывести сообщение об ошибке.
2. Написать программу, создающую вектор чисел, и копирующую все числа, кратные заданному числу, в новый вектор.
3. На одометре (прибор для измерения пробега автомобиля) число 15951. Через 2 часа езды он показывает другое число палиндром. Какова средняя скорость? Написать программу, позволяющую задать новое значение на одометре (новое число должно быть больше начального (15951) и быть палиндромом) и вычисляющую среднюю скорость.

# 1 Основная часть

## 1.1 Общая идея решения задачи

Для решения задачи были использованы:

1. Алгоритмы стандартной библиотеки `<algorithm>`: *remove*, *search*, *generate*, *copy if*, *reverse copy*, *for each*.
2. Контейнеры `<vector>` и `<string>`.
3. Функция *srand* из библиотеки `<cstdlib>` и функция *time* из библиотеки `<ctime>` для генерации случайных чисел.
4. Лямбда-выражения.

## 1.2 Структура и принципы действия

### 1.2.1 Задание 1

В функции *task 5* происходит считывание произвольной строки, введенной с клавиатуры. Далее с помощью функции *remove* удаляются все пробелы из строки. Затем функция *erase* позволяет удалить из строки ненужные элементы, появившиеся после исключения всех пробелов. После происходит считывание подстроки, функция *search* проверяет входит ли эта последовательность символов в строку и возвращает итератор к первому элементу вхождения. Если итератор указывает на конец строки, то выводится сообщение об ошибке, то есть подстрока не найдена.

### 1.2.2 Задание 2

В функции *task 11* происходит создание двух векторов одинаковой длины (для наглядности длина 10). Далее инициализируется генератор случайных чисел с помощью функций *srand* и *time*. Затем функция *generate* заполняет вектор случайными значениями (для удобства в диапазоне от 0 до 9), используя функцию-генератор в виде лямбда-выражения. После этого пользователь вводит число, чтобы проверить значения вектора на кратность используется функция *copy if*. Она копирует значения из одного вектора в другой, если параметр-функция, реализованная в виде лямбда-выражения, вернет *true*. Лямбда-выражение проверяет число из вектора на кратность введенному числу.

### 1.2.3 Задание 3

В функции *task 14* осуществляется генерация случайного вектора-палиндрома. Для этого генератор случайных чисел инициализируется с помощью функций *srand* и *time*. Затем функция *generate* заполняет вектор длины 5 случайными значениями от 0 до 9, используя функцию-генератор в виде лямбда-выражения. Далее функция *reverse copy* отображает вектор в обратном порядке. Операции генерации и отображения повторяются, пока не будет создан вектор-палиндром, больший заданного в условии вектора. После того, как вектор будет получен, с помощью функции *for each* и лямбда-выражения вектор переводится в число, и затем считается и выводится на экран средняя скорость.

## 1.3 Процедура получения исполняемых программных модулей

Программный код был скомпилирован с среде *Visual Studio 2017*. Код программы содержится в одном исходном файле. Для ускоренной компиляции программы используются предварительно откомпилированные заголовки *"pch.h"*. Помимо этого никаких дополнительных ключей не добавлялось, использовались ключи, которые добавляются по умолчанию.

## 1.4 Результаты тестирования

Тестирование программы представлено в файле *"Task5.cpp"* в функции *Main()*. Ожидаемый вывод функции:

```
your string:microsoft visual studio top
new string:microsoftvisualstudiotop
your substring:nottop
Can't find
```

```
random vector: 7 4 2 7 0 8 5 2 9 6
your number: 7
new vector: 7 7 0
```

```
new value: 95859
speed: 39954
```

# Приложение А

полный код программы

## A.1 - Task5.cpp

```
#include "pch.h"
#include <iostream>
#include <string>
#include <algorithm>
#include <vector>
#include <ctime>

using namespace std;

void task5() {
    string s, s1;
    cout << "your_string:";
    getline(cin, s);
    auto pos = remove(s.begin(), s.end(), ' ');
    s.erase(pos, s.end());
    cout << "new_string:" << s;
    getchar();
    cout << "your_substring:";
    getline(cin, s1);
    auto pos1 = search(s.begin(), s.end(),
                      s1.begin(), s1.end());
    if (pos1 == s.end())
        cerr << "Can't find" << endl;
}

void task11() {
    vector<int> v(10), v1(v.size());
    int i;
    srand(time(NULL));
    generate(v.begin(), v.end(),
            []() { return rand()%10; });
```

```

    cout << "random_vector:";
    for (int& x : v) cout << ' ' << x;
    cout << endl << "your_number:";
    cin >> i;
    auto it = copy_if(v.begin(), v.end(),
                     v1.begin(),
                     [i](int e){return (e % i == 0);});
    v1.erase(it, v1.end());
    cout << "new_vector:";
    for (int& x : v1) cout << ' ' << x;
    cout << endl;
}

void task14() {
    srand(time(NULL));
    vector<int> x(5), x1(5), y = {1,5,9,5,1};
    do{
        generate(x.begin(), x.end(),
                []() {return rand() % 10;});
        reverse_copy(x.begin(), x.end(),
                    x1.begin());
    } while (x < y || x != x1);
    cout << "new_value:";
    for (int& i : x) cout << i;
    float val = 0;
    for_each(x.begin(), x.end(),
            [&val](int elm) { val = val * 10 + elm; });
    cout << endl << "speed:" << (val - 15951) / 2;
}

int main(){
    task5();
    task11();
    task14();
    return 0;
}

```