МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ПО ИНДИВИДУАЛЬНОМУ ЗАДАНИЮ № 3**

**по дисциплине  
 «Параллельное и низкоуровневое программирование»**

Выполнил студент группы 25/2                                       А.А. Козин

Направление подготовки  02.03.03  Математическое обеспечение и администрирование информационных систем

Курс    2

Отчет принял доктор физико-математических наук, профессор                                                                                       А.И. Миков

Краснодар

2022 г.

**Задание 1 Целочисленная арифметика**: Написать программу, вычисляющую «следующее простое число» по заданному простому числу n, т.е. находящую такое простое число k > n, что все числа между k и n – составные.  
**Решение**: Для решение поставленной задачи было использована 1 ассемблерная вставка и 2 функции.   
Функция isPrime проверяет число на простоту, а isComposite проверят составные числа.  
Ассемблерная вставка используется для присваивания переменной **k** значения **n**, а далее прибавляет единицу к **k**. Прибавляется единица к **k** для того, чтобы начать идти с составного числа до следующего простого т.к. если не прибавить единицу **for** найдет тоже самое простое число и выйдет из цикла. Так же из-за действий описанных выше k > n будет всегда.

**Текст программы:**

#include <iostream>

#include <Windows.h>

using namespace std;

bool isPrime(int n) {

if (n <= 1)

return false;

for (int i = 2; i < n; i++)

if (n % i == 0)

return false;

return true;

}

bool isComposite(int k) {

for (int i = 2; i < k; i++)

if (k % i == 0)

return true;

return false;

}

void main() {

int n, k;

cout << "Enter the Number to check Prime: "; cin >> n;

if (isPrime(n)) {

cout << n << " is a prime number" << endl << endl;

\_\_asm {

mov eax, n;

mov ebx, k;

mov ebx, eax;

add ebx, 1;

mov k, ebx;

}

for (k; k <= 10000; k++) {

if (isComposite(k)) {

cout << k << " is a Composite number" << endl;

}

else {

cout << endl << "Next prime number " << k << endl;

cout << "Interval between prime numbers " << "[" << n << "; " << k << "]" << endl;

break;

}

}

}

else

cout << n << " is NOT a prime number" << endl;

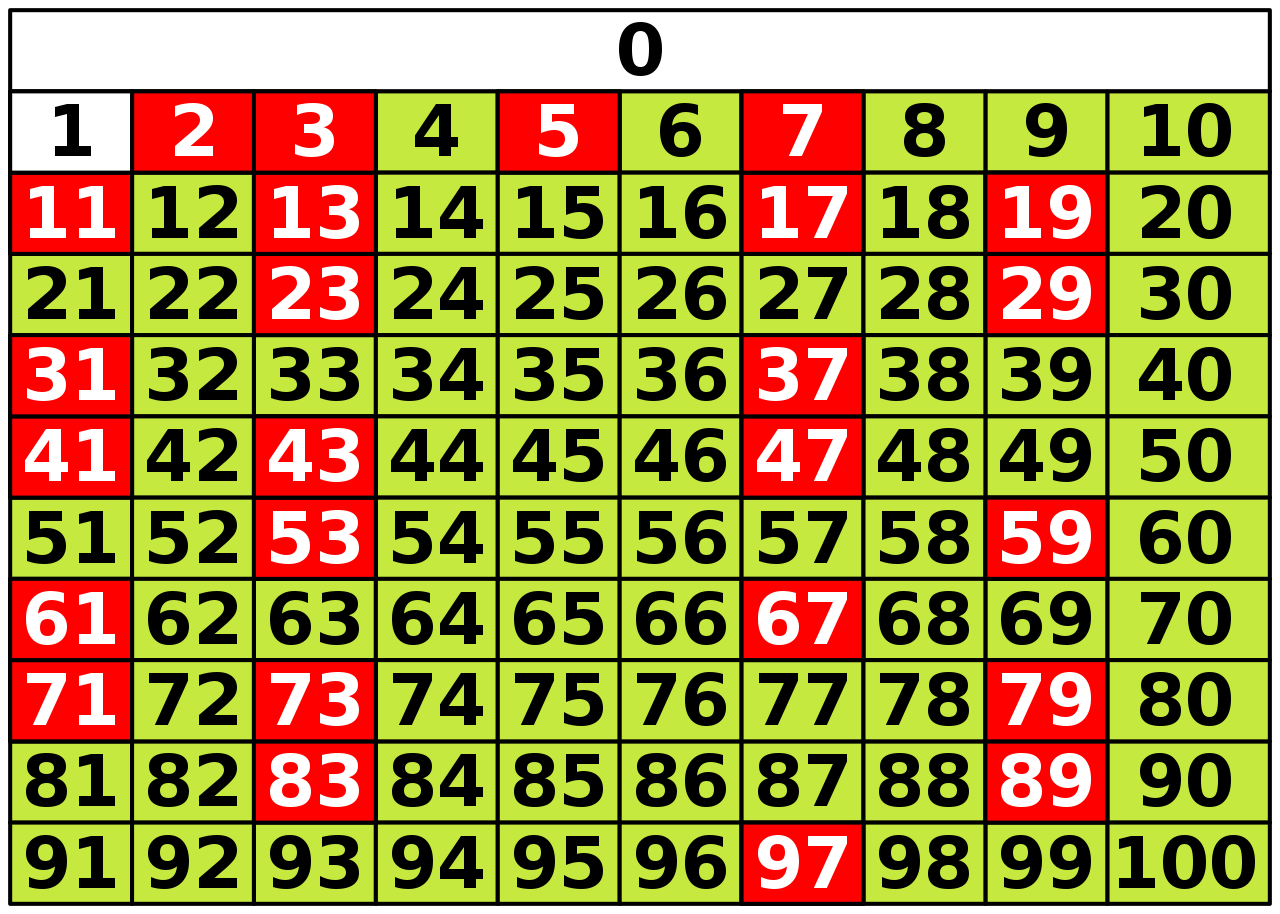
}

**Программа выдает в консоли результаты в следующем виде:**

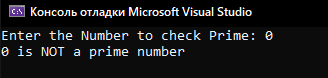
В первую очередь в консоли выводиться результат проверки введённого числа на простоту.

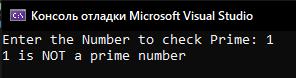
Если число простое, то далее будут выводиться составные числа. До того момента пока не найдется первое простое число большее заданного и ниже выводиться интервал между простыми числами.

Иначе, в консоли будет выведено, что число непростое.

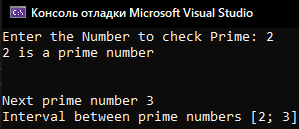
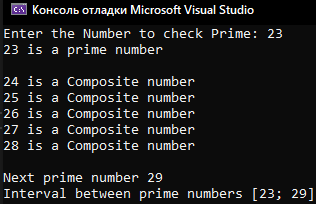
Красные числа простые, а зеленые составные.

Test 1:





Test 2:



Test 3:

