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 вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Криптографические протоколы**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Иванищев А.А.

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Крамаренко А. А.

**Индивидуальное задание.**

Реализовать программный продукт нахождения всех первообразных корней по заданному простому модулю с указанием всех этапов нахождения корня.

Первообразным корнем по модулю m называется целое число g, такое, что



и



где – функция Эйлера.

**Ход работы.**

В ходе работы была разработана программа, которая находит первообразные корни заданного числа. На первом этапе находится количество первообразных корней для данного числа с помощью формулы или для простого числа (Теорема о количестве первообразных корней).

Вторым этапом определим числа в каких степенях будем проверять. Кажется, что мы должны перебрать все степени меньше , но на самом деле алгоритм можно оптимизировать.

Из теоремы Лагранжа следует, что показатель любого числа по модулю n является делителем \phi(n). Таким образом, достаточно проверить, что для всех собственных делителей d\ |\ \phi(n) выполняется g^d \not\equiv 1 \pmod{n}. Это уже значительно более быстрый алгоритм, однако можно пойти ещё дальше.

Факторизуем число \phi(n) = p_1^{a_1} \ldots p_s^{a_s}. Докажем, что в предыдущем алгоритме достаточно рассматривать в качестве d лишь числа вида \frac{ \phi(n) }{ p_i }. Действительно, пусть d — произвольный собственный делитель \phi(n). Тогда, очевидно, найдётся такое j, что d\ |\ \frac{ \phi(n) }{ p_j }, т.е. d \cdot k = \frac{ \phi(n) }{ p_j }. Однако, если бы g^d \equiv 1 \pmod{n}, то мы получили бы:

 g^{\frac{ \phi(n) }{ p_j }} \equiv g^{d \cdot k} [...]

т.е. всё равно среди чисел вида \frac{ \phi(n) }{ p_i } нашлось бы то, для которого условие не выполнилось, что и требовалось доказать. Таким образом, алгоритм нахождения первообразного корня такой. Находим \phi(n), факторизуем его. Теперь перебираем все числа g = 1 \ldots n, и для каждого считаем все величины g^{ \frac{ \phi(n) }{ p_i } } \pmod{n}. Если для текущего g все эти числа оказались отличными от 1, то это g и является искомым первообразным корнем.

Поэтому на втором этапе определяем все степени, в которые будем возводить взаимно простые числа меньшие .

На дальнейших этапах определяем подходит ли нам корень или нет. Подробная работа программы для чисел 11 и 17.

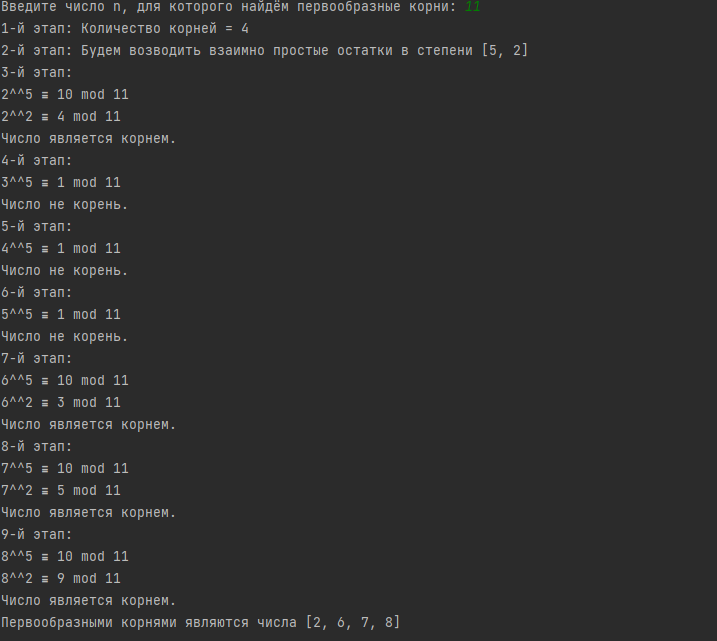
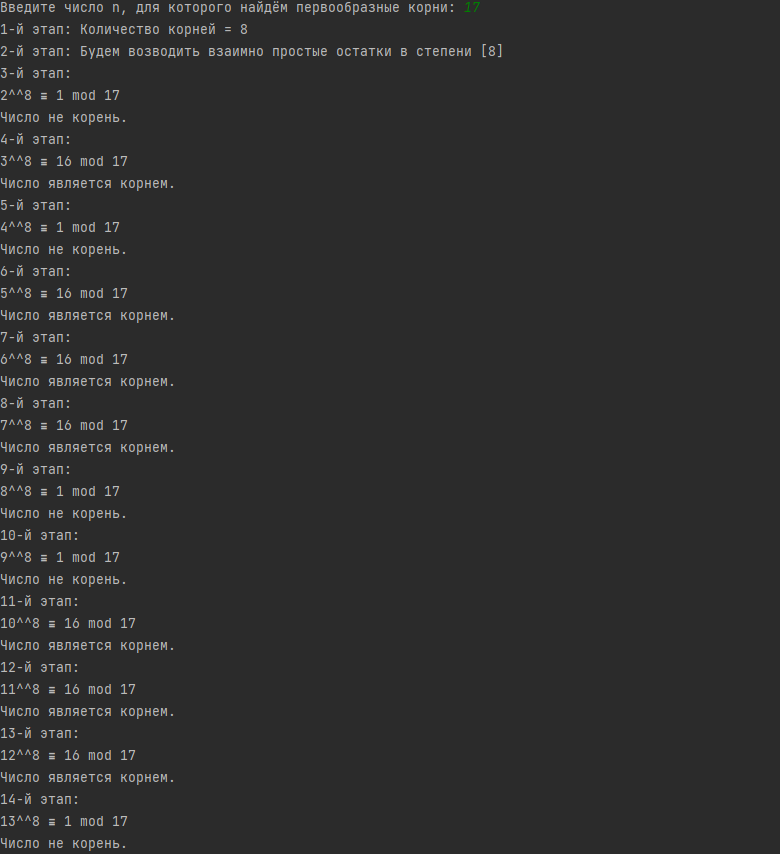
**

Рисунок 1 – работа программы для введённого числа 11.



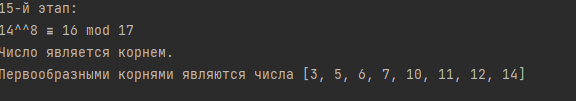
****

Рисунок 2 – работа программы для числа 17.