using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Numerics;

namespace RSA

{

class Program

{

static char[] characters = new char[] { '#', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И',

'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С',

'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ь', 'Ы', 'Ъ',

'Э', 'Ю', 'Я', ' ', '1', '2', '3', '4', '5', '6', '7',

'8', '9', '0', ',', '.', '-', '!', '?',

'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I',

'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',

'U', 'V', 'W', 'X', 'Y', 'Z'};

//символы зашифровки и расшифровки

static long d, n;

static List<string> result; //результат для блоков

static void Main(string[] args) //главная функция

{

Repeat:

Console.WriteLine("Алгоритм RSA");

Console.WriteLine("Введите большое простое число p: ");

string P = Console.ReadLine();

Console.WriteLine("Введите большое простое число q, с такой же длинной как число p: ");

string Q = Console.ReadLine();

Console.WriteLine("Введите сообщение: ");

string S = Console.ReadLine();

if (P.Length > 0 && Q.Length > 0) //проверка на длину консольного ввода

{

long p = Convert.ToInt64(P);

long q = Convert.ToInt64(Q);

if(IsTheNumberSimple(p) && IsTheNumberSimple(q)) //проверка на простоту введеных чисел

{

string s = S.ToUpper();

n = p \* q;

long m = (p - 1) \* (q - 1);

d = Calculate\_d(m);

long e\_ = Calculate\_e(d, m);

result = RSA\_Endoce(s, e\_, n);

string res = "";

foreach(string item in result)

{

res += item + " ";

}

Console.WriteLine($"Блоки сообщения: {res}");

Console.WriteLine($"Секретный ключ d: {d}");

Console.WriteLine($"Секретный ключ n: {n}");

} else

{

Console.WriteLine("p и q - не простые числа!");

goto Repeat;

}

} else

{

Console.WriteLine("Введите p и q");

goto Repeat;

}

Part2:

Console.WriteLine("Наберите 1 - Расшифровать, 2 - Начать сначала 0 - Выйти из программы");

string check = Console.ReadLine();

if (check == "1") //проверка на введеный символ

{

List<string> input = result;

string res = RSA\_Dedoce(input, d, n);

Console.WriteLine($"Расшифрованное сообщение: {res}");

goto Part2;

} else if (check == "2")

{

goto Repeat;

} else if (check == "0")

{

return;

}

}

//проверка на простоту числа

private static bool IsTheNumberSimple(long n)

{

if (n < 2)

return false;

if (n == 2)

return true;

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

if (n % i == 0)

return false;

return true;

}

//зашифровать

private static List<string> RSA\_Endoce(string s, long e, long n)

{

List<string> result = new List<string>();

BigInteger bi;

for (int i = 0; i < s.Length; i++)

{

int index = Array.IndexOf(characters, s[i]);

bi = new BigInteger(index);

bi = BigInteger.Pow(bi, (int)e);

BigInteger n\_ = new BigInteger((int)n);

bi = bi % n\_;

result.Add(bi.ToString());

}

return result;

}

//расшифровать

private static string RSA\_Dedoce(List<string> input, long d, long n)

{

string result = "";

BigInteger bi;

foreach (string item in input)

{

bi = new BigInteger(Convert.ToDouble(item));

bi = BigInteger.Pow(bi, (int)d);

BigInteger n\_ = new BigInteger((int)n);

bi = bi % n\_;

int index = Convert.ToInt32(bi.ToString());

result += characters[index].ToString();

}

return result;

}

//вычисление параметра d. d должно быть взаимно простым с m

private static long Calculate\_d(long m)

{

long d = m - 1;

for (long i = 2; i <= m; i++)

if ((m % i == 0) && (d % i == 0)) //если имеют общие делители

{

d--;

i = 1;

}

return d;

}

//вычисление параметра e

private static long Calculate\_e(long d, long m)

{

long e = 10;

while (true)

{

if ((e \* d) % m == 1)

break;

else

e++;

}

return e;

}

}

}