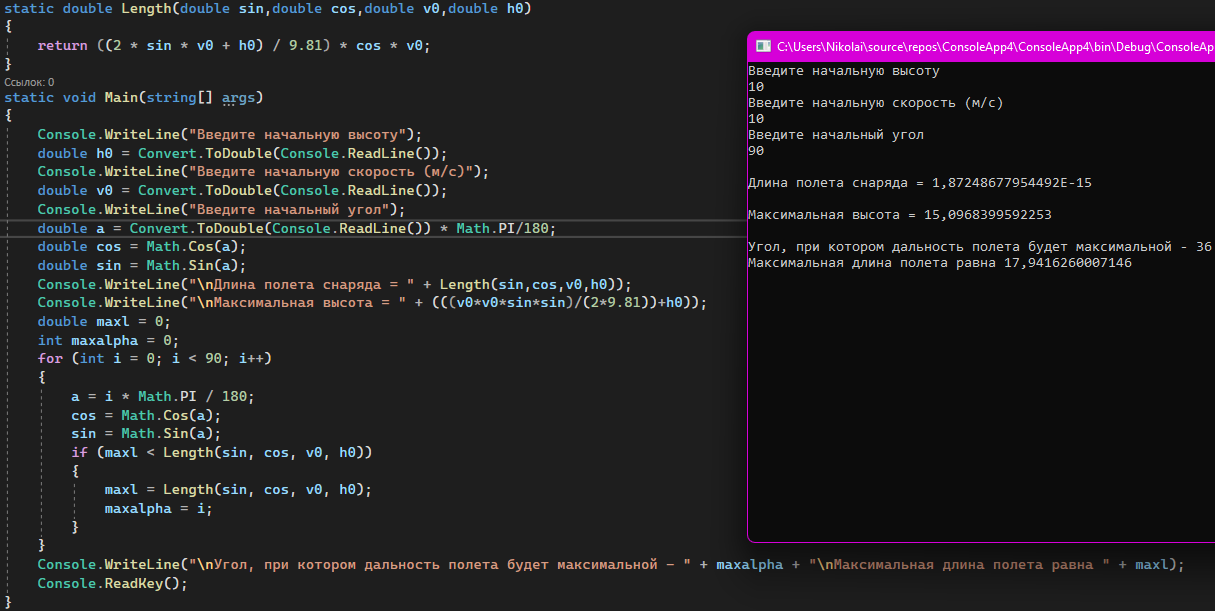
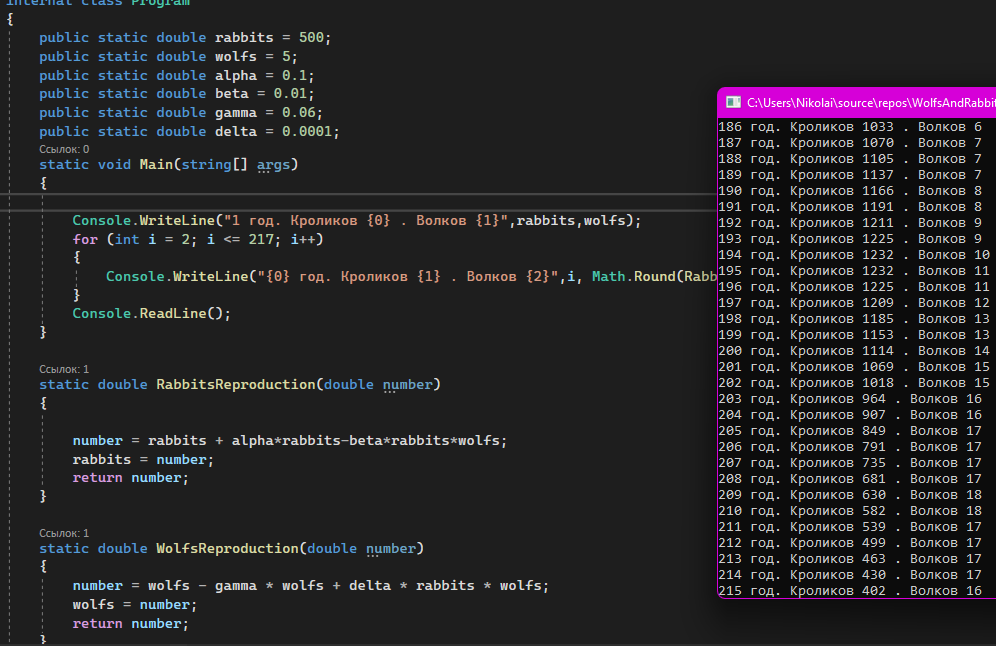
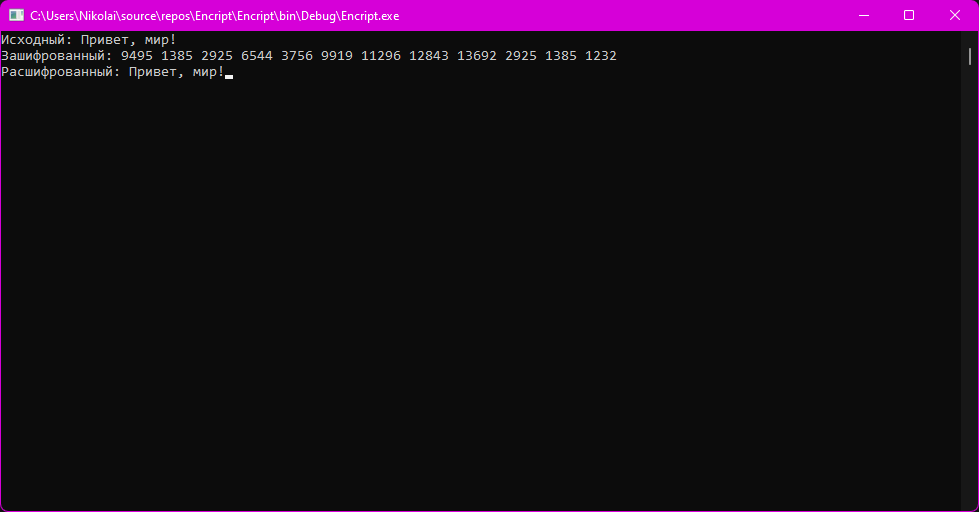
Отчет по проделанной работе

1. Задача на бросок под углом
2. Задача на кроликов и волков
3. Задача на шифрование и расширование

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Encript

{

internal class Program

{

static int isSimple(int number)

{

if (number == 1)

return 0;

for (int i = 2; i <= Math.Sqrt(number); i++)

{

if ((number % i == 0))

return 0;

}

return 1;

}

static int nSimple(int n)

{

int i = 0, nuber = 0;

while (i < n)

{

nuber++;

i += isSimple(nuber);

}

return nuber;

}

static int ostat(int x, int y, long z)

{

int ost = x;

for (int i = y - 1; i > 0; i--)

{

ost = (ost \* x) % Convert.ToInt32(z);

}

return ost;

}

static int closeKey(int p, int q, int e)

{

ulong fi = Convert.ToUInt32((p - 1) \* (q - 1));

int d = 0;

ulong ost = 0;

while (ost != 1)

{

d++;

if (isSimple(d) == 1)

{

ost = Convert.ToUInt32((e \* d)) % fi;

}

}

return d;

}

static List<int> code (char[] c, int e, long n)

{

List<int> rez = new List<int>();

for (uint i = 0; i < c.Length; i++)

{

rez.Add((char)ostat(c[i], e, n));

}

return rez;

}

static List<char> decode(List<int> c, long d, long n)

{

List<char> rez = new List<char>();

for (int i = 0; i < c.Count; i++)

{

rez.Add((char)ostat(c[i], (int)d, n));

}

return rez;

}

static void Main(string[] args)

{

Random random = new Random();

int p = nSimple(29);

int q = nSimple(31);

long n = p \* q;

int e = 257;

long d = closeKey(p, q, e);

int symbol = 666;

int shifr = ostat(symbol, e, n);

char[] phraze = "Привет, мир!".ToCharArray();

Console.Write("Исходный: ");

foreach (char c in phraze)

{

Console.Write(c);

}

Console.WriteLine();

List<int> codePhraze = code(phraze, e, n);

List<char> decoding = decode(codePhraze, d, n);

Console.Write("Зашифрованный: ");

foreach (int item in codePhraze)

{

Console.Write($"{item} ");

}

Console.WriteLine();

Console.Write("Расшифрованный: ");

foreach (char decode in decoding)

{

Console.Write(decode);

}

Console.ReadLine();

}

}

}