

## Курсовой проект.

№0. Строку в которой записано своё ФИО подать на вход в хеш-функцию ГОСТ Р 34.11-2012 (Стрибог). Младшие 4 бита выхода интерпретировать как 16-тиричное число, которое в дальнейшем будет номером варианта.

№1. Программно реализовать один из алгоритмов функции хеширования в соответствии с номером варианта. Алгоритм содержит в себе несколько раундов.

№2. Модифицировать оригинальный алгоритм таким образом, чтобы количество раундов было настраиваемым параметром программы. в этом случае новый алгоритм не будет являться стандартом, но будет интересен для исследования.

№3. Применить подходы дифференциального криптоанализа к полученным алгоритмам с разным числом раундов.

№4. Построить график зависимости количества раундов и возможности различения отдельных бит при количестве раундов 1,2,3,4,5,... .

№5. Сделать выводы.

Примечание №1. Допустимо использовать сторонние реализации для пункта 1, при условии, что они проходят тесты из стандарта и пригодны для дальнейшей модификации.

Примечание №2. Если в алгоритме описывается семейство с разными размерами блоков, то можно выбрать любой из них.

### Приложение №1.

Номер варианта == Алгоритм

0 == ГОСТ Р 34.11-94

1 == ГОСТ Р 34.11-2012 (Стрибог)

2 == Luffa

3 == BLAKE

4 == SHA-0

5 == SHA-1

6 == SHA-2

7 == Кеccak

8 == JH

9 == Shabal

A == Skein

B == Blue Midnight Wish-256

C == CubeHash

D == MD5

E == SIMD

F == Whirlpool

### Приложение №2.

Процесс выбора варианта, также требуется отразить в отчёте.

В примере ниже описан выбор варианта с помощью библиотеки pygost.

```
from pygost import gost34112012256
gost34112012256.new("Иванов Иван Иванович").digest();
'\xe6\xe1\xf5H\x94\xe8\xeah@\xe6Pl\xa4&\xfb\x12za-A\xa6\x08\xc2m\xfe\xfbL[\x94N4\xbe';
# => вариант 'e'
gost34112012256.new("Сидоров Иван Иванович").digest();
'\xba\xb38\x82\xc6\x95>e@c\x15%\xfc]\xca\x97X\x19h\xd9\xd3C\xcf\xcci\xa7*R{\x84+\xc8'
# => вариант '8'
```