

1 Тестирование hash

1.1 default

Входные данные – MD5-хеш строки *hello*, то есть *5d41402abc4b2a76b9719d911017c592*.

Ожидаемый результат:

- `res == "hello"`.

1.2 short_hash

Входные данные – строка длиной в 31 символ, что меньше, чем длина MD5-хеша.

Ожидаемый результат:

- `res == "Это не MD5 хэш!"`.

1.3 long_hash

Входные данные – строка длиной в 33 символа, что больше, чем длина MD5-хеша.

Ожидаемый результат:

- `res == "Это не MD5 хэш!"`.

1.4 nothing_find

Входные данные – MD5-хеш строки, которой нет в словаре (*rockyou.txt*).

Ожидаемый результат:

- `res == "Ничего не нашлось :("`.

2 Тестирование rsa

2.1 prime_p

Входные данные – $p = 62$, $q = 53$, $e = 17$, $ct = 3$. p – непростое число.

Ожидаемый результат:

- `res == -1`.

2.2 prime_q

Входные данные – $p = 61$, $q = 52$, $e = 17$, $ct = 3$. q – непростое число.

Ожидаемый результат:

- `res == -1`.

2.3 prime_e

Входные данные – $p = 61$, $q = 53$, $e = 10$, $ct = 3$. e – непростое число.

Ожидаемый результат:

- `res == -1`.

2.4 example

Входные данные – $p = 61$, $q = 53$, $e = 17$, $ct = 2790$. Пример из Википедии.

Ожидаемый результат:

- `res == 65`.

2.5 zero

Входные данные – $p = 0$, $q = 0$, $e = 0$, $ct = 10$. Проверка на нулевые аргументы.
Ожидаемый результат:

- `res == -1`.

2.6 prime

Входные данные – $n = 10$, $n = 11$, $n = -10$. Проверка на простые числа. Функция `IsPrime`.
Ожидаемый результат:

- `res == false`, `res2 == true`, `res3 == false`

2.7 coprime_d

Входные данные – $e = 6$, $\phi = 10$. Проверка на расчёт сопростого числа. Функция `calculateD`.
Ожидаемый результат:

- `res == -1`

2.8 coprime_d_zero

Входные данные – $e = 0$, $\phi = -10$. Проверка на расчёт сопростого числа. Функция `calculateD`.
Ожидаемый результат:

- `res == -1`

3 Тестирование bases

3.1 example

Входные данные – строка *lalala*.
Ожидаемый результат:

- `res[0] == "6C616C616C61"`
- `res[1] == "NRQWYYLMME=====`
- `res[2] == "bGFsYWxh"`