Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №3
“SHA-3”

Выполнил:
студент группы ИУ5-35Б:
Купцов С.Р.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.В.
Подпись и дата:

Москва, 2024 г.

Задание

Реализовать алгоритм keccak на языке python

Код программы

SHA-3

```
from bitstring import Bits

r = 1024
rounds = 24


def round(A, r):
    # θ
    A1 = [[[0 for i in range(64)] for j in range(5)] for k in range(5)]
    for i in range(5):
        for j in range(5):
            for k in range(64):
                C = sum([A[(i - 1) % 5][h][k] for h in range(5)]) % 2
                D = sum([A[((i + 1) % 5)][h][(k - 1) % 64] for h in
range(5)]) % 2
                temp = C + D + A[i][j][k] % 2
                A1[i][j][k] = temp
    # ρ
    r_ = [[0, 36, 3, 41, 18], [1, 44, 10, 45, 2], [62, 6, 43, 15, 61], [28,
55, 25, 21, 56], [27, 20, 39, 8, 14]]
    A = [[[0 for i in range(64)] for j in range(5)] for k in range(5)]
    for i in range(5):
        for j in range(5):
            for k in range(64):
                A[i][j][k] = A1[i][j][k - r_[i][j]]
    # π
    A1 = [[[0 for i in range(64)] for j in range(5)] for k in range(5)]
    for i in range(5):
        for j in range(5):
            for k in range(64):
                A1[j][(2 * i + 3 * j) % 5][k] = A[i][j][k]
    # χ
    A = [[[0 for i in range(64)] for j in range(5)] for k in range(5)]
    for i in range(5):
        for j in range(5):
            for k in range(64):
                A[i][j][k] = (A1[i][j][k] + (((A1[(i + 1) % 5][j][k] + 1) %
2) * (A1[(i + 2) % 5][j][k]))) % 2
    # ι
    rc = [0] * 168

    w = [1, 0, 0, 0, 0, 0, 0, 0, 0]
    rc[0] = 1
    for i in range(1, 168):
        w = [w[1], w[2], w[3], w[4], w[5], w[6], w[7], (w[0] + w[4] + w[5] +
w[6]) % 2]
        rc[i] = w[0]

    for l in range(7):
        A[0][0][2 ** l - 1] ^= rc[l + 7 * r]
    return A
```

```
def f(RC, b_text):
    for x in range(5):
        for y in range(5):
            for z in range(64):
                if 64 * (5 * y + x) + z < 1024:
                    RC[x][y][z] = int(b_text[64 * (5 * y + x) + z])
                else:
                    RC[x][y][z] = 0

    for i in range(rounds):
        RC = round(RC, i)
    return RC


def SHA3(text):
    RC = [[[0 for i in range(64)] for j in range(5)] for k in range(5)]

    b_text = Bits(bytes=bytes(text, encoding='utf-8')).bin
    b_text += '1'
    if len(b_text) % r != 0:
        b_text += '0' * (r - len(b_text) % r - 1)
    b_text += '1'

    while b_text:
        RC = f(RC, b_text[:1024])
        b_text = b_text[1024:]

    hash = ''
    for z in range(64):
        for x in range(5):
            for y in range(5):
                hash += str(RC[x][y][z])
    return Bits(bin=hash).hex[:256]


while True:
    print(SHA3(input()))
```

Экранные формы с примерами выполнения программы