

Санкт-Петербургский Национальный Исследовательский Университет
Информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Домашняя работа №2
Вариант №8

Выполнили:
Смирнов И.И.
Телунц Э.Р.
Царев А.С.
Проверил:
Мусаев А.А.

Санкт-Петербург
2022

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Задание 1	4
2 Задание 2	5
ЗАКЛЮЧЕНИЕ	6
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	7

ВВЕДЕНИЕ

Для становления хорошим специалистом в области программирования на языке Python необходимо знать основные алгоритмы и функционал языка.

Цель данной работы – ознакомление с хэш-функциями и их реализация на языке Python.

В ходе лабораторной работы были решены следующие задачи:

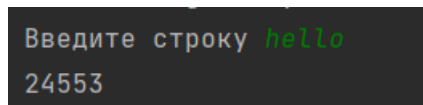
- написание алгоритма хэширования по методу умножения;
- написание алгоритма хэширования по методу md5.

Задания, которые необходимо выполнить:

- 1) написать алгоритм, который осуществляет хэширование методом умножения введенной пользователем строки;
- 2) написать алгоритм, который осуществляет хэширование методом md5 введенной пользователем строки.

1 Задание 1

В данном задании необходимо произвести хэширование строки по методу умножения. Для этого после получения строки от пользователя каждый символ строки переводится в код согласно таблице ASCII и этот код добавляется в массив. Далее для каждого такого кода применяется формула $M * ((Key * C) \bmod 1)$ с округлением по математическим правилам, где Key - код символа, C - константа из диапазона $[0, 1]$, M - длина строки. Каждый полученный результат добавляется в массив. Когда все коды преобразованы по данной формуле, результаты склеиваются в одну строку. Числовое значение этой строки и является хэшем.

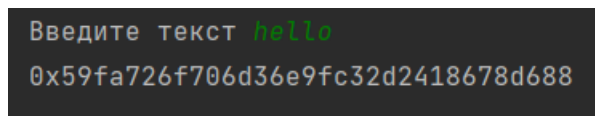


```
Введите строку hello
24553
```

Рисунок 1 - Вывод алгоритма хэширования по методу умножения

2 Задание 2

В данном задании необходимо произвести хэширование строки от пользователя по методу md5. Для начала полученное от пользователя сообщение переводится в 16-байтный формат и выравнивается так, чтобы длина сообщения была сравнима с 448 по модулю 512. После этого в конец дописывается 64 битное представление длины сообщения. Далее инициализируются 4 буфера и 4 вспомогательных бинарных функции. Далее для всех значений i в диапазоне $[1, 64]$ вычитывается шумовая составляющая по формуле $T = 4294967296 * |\sin(i)|$. Также происходит разделение строки на подстроки длиной 512 бит, которые будут разбиты на блоки по 32 бита. После этого производятся необходимые вычисления хэша с использованием ранее обозначенных констант, буферов и шумовых составляющих в 4 этапа. Результат всех этих вычислений - 4 изменных буфера. После их перевода в 16-ричную систему и склеивания как строк, получается хэш заданного сообщения.



```
Введите текст hello
0x59fa726f706d36e9fc32d2418678d688
```

Рисунок 2 - Вывод алгоритма хэширования md5

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе был получен опыт работы на языке Python, а также понимание работы алгоритмов хэширования, а именно метода умножения и md5.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Wikipedia: официальный сайт: <https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F> (Дата обращения 06.03.2023)

Ссылка на полный код

<https://github.com/Igor2551/Homework2>