Лабораторная работа 1

Терентьев Егор Дмитриевич, НФИмд-01-23

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

ПРЕЗЕНТАЦИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

дисциплина: Математические основы защиты информации и информационной безопасности

Преподователь: Кулябов Дмитрий Сергеевич
Студент: Терентьев Егор Дмитриевич
Группа: НФИмд-01-23

МОСКВА

2022 г.

Прагматика выполнения лабораторной работы

Прагматика выполнения лабораторной работы

Требуется реализовать:

1. Шифр Цезаря с произвольным ключом K.
2. Шифр Атбаш.

Цель работы

Цель работы

Освоить на практике шифры простой замены.

Выполнение лабораторной работы

1. Для реализации шифра цезаря создал функции:

1. Для реализации шифра цезаря создал функции:

1. проверяющая правильность введенного ключа
2. проверяющая введенное значения для пароля и слова для шифрования
3. добавляющая только уникальные буквы из слова

```
def is_key(key):
    try:
        key = int(key)
        if key > len(alphabet):
            key %= len(alphabet)
        return key
    except ValueError:
        print("key is not int")
        raise ValueError

def is_correct_input_word_value(word, alphabet):
    for letter in word:
        if letter not in alphabet:
            print("Incorrect input word value")
            raise ValueError

def unique_letters(password):
    unique = []
    for letter in password:
```

2. Также еще нужны функции создания шифр-алфавита и шифрования слова

2. Также еще нужны функции создания шифр-алфавита и шифрования слова

```python
def create_code_alpabet(alphabet, uniquePassLetters, key):
    codeAlphabet = []
    alphabet_without_upl = []
    for letter in alphabet:
        if letter not in uniquePassLetters:
            alphabet_without_upl.append(letter)

    codeAlphabet = alphabet_without_upl[-key:]
    codeAlphabet += uniquePassLetters
    codeAlphabet += alphabet_without_upl[:-key]
    return codeAlphabet

def word_to_code(word, alphabet, codeAlphabet):
    codeWord = ""
    for letter in word:
        for i in range(len(alphabet)):
            if alphabet[i] == letter:
                codeWord += codeAlphabet[i]
                break
    return codeWord
```

Figure 2: caesar_func_2

3. Основная фунция запуска где получаем входные значения и шифруем слово

3. Основная фунция запуска где получаем входные значения и шифруем слово

```
44
45  alphabet = ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п',
46              'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']
47  key = input("key: ")
48  key = is_key(key=key)
49  password = input("Password: ").lower()
50  is_correct_input_word_value(word=password, alphabet=alphabet)
51  uniquePassLetters = unique_letters(password=password)
52  codeAlphabet = create_code_alpabet(alphabet=alphabet, uniquePassLetters=uniquePassLetters, key=key)
53
54  print(alphabet)
55  print(codeAlphabet)
56
57  wordToCode = input("input word to code: ")
58  is_correct_input_word_value(wordToCode, alphabet)
59  codeWord = word_to_code(word=wordToCode, alphabet=alphabet, codeAlphabet=codeAlphabet)
60  print("new coded word: " + codeWord)
```

Figure 3: caesar_main_func

4. Запуск программы

4. Запуск программы

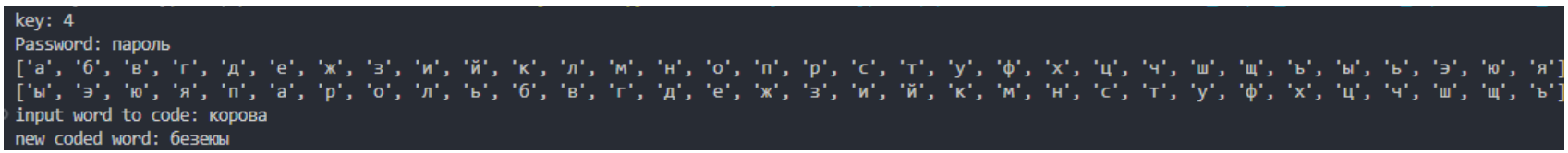

Figure 4: output_caesar

5. Для реализации шифра атбаш создал функции:

5. Для реализации шифра атбаш создал функции:

1. для проверки что каждая буква входит в алфавит (для входных значений)
2. разворот алфавита
3. кодировка слова с помощью шифр-алфавита

```
def are_letters_from_alphabet(password):
    for letter in password:
        if letter not in alphabet:
            print("Incorrect input value")
            raise ValueError

def atbash_reverse(alphabet):
    return alphabet[::-1]

def word_to_code(word, alphabet, codeAlphabet):
    codeWord = ""
    for letter in word:
        for i in range(len(alphabet)):
            if alphabet[i] == letter:
                codeWord += codeAlphabet[i]
```

6. Основная фунция запуска шифра Атбаш, где получаем входные значения и шифруем слово

6. Основная фунция запуска шифра Атбаш, где получаем входные значения и шифруем слово

```
alphabet = ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п',
            'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я', ' ']
atbashAlphabet = atbash_reverse(alphabet=alphabet)

print(alphabet, "\n", atbashAlphabet)

wordToCode = input("input word to code: ")
are_letters_from_alphabet(wordToCode)

codedword = word_to_code(word=wordToCode, alphabet=alphabet, codeAlphabet=atbashAlphabet)
print("coded word: " + codedword)
```

Figure 6: atbash_main_func

7. Запуск программы атбаш

7. Запуск программы атбаш

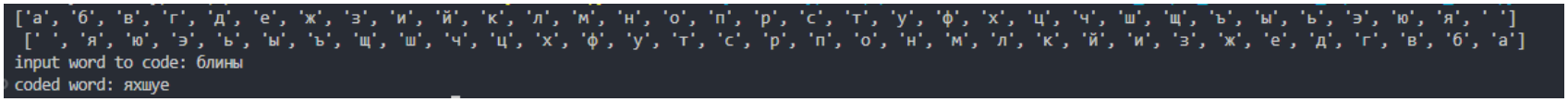

Figure 7: output_atbash

Выводы

Выводы

В результате выполнения работы я освоил на практике применение шифров простой замены.