

Шифрование гаммированием

Гаджиев Нурсултан Тофик оглы

2022 Moscow, Russia

RUDN University, Moscow, Russian Federation

Цель работы

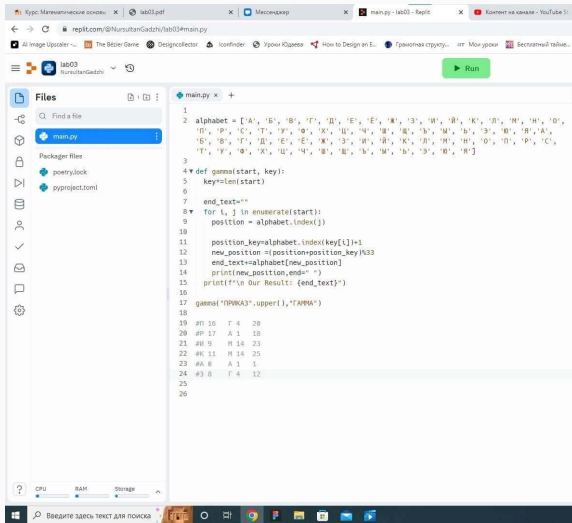
Реализация алгоритма шифрования гаммированием конечной гаммой.

Задачи

1. Реализовать алгоритм шифрования гаммированием конечной гаммой.

Реализация

Функция гамма для шифрования текста. (рис. 1)

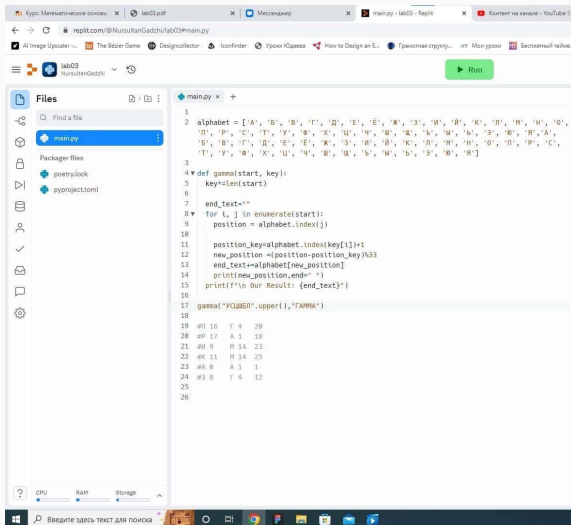


The screenshot shows a web-based Python IDE interface. The left sidebar displays a file explorer with a folder named 'lab03' containing a file 'main.py'. The main editor area shows the code for the 'gamma' function. The code defines an alphabet and a function that takes a start string and a key, then iterates through the start string to produce a new string by shifting characters in the alphabet based on the key. The output of the function is printed as 'Our Result: {end_text}'.

```
1
2 alphabet = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
3
4 def gamma(start, key):
5     key = len(start)
6
7     end_text = ""
8     for i, j in enumerate(start):
9         position = alphabet.index(j)
10
11         position_key = alphabet.index(key[i]) + 1
12         new_position = (position + position_key) % 26
13         end_text = alphabet[new_position]
14         print(new_position, end=" ")
15     print(f"\n Our Result: {end_text}")
16
17 gamma("ПРИКАЗ", upper(), "ГММА")
18
19 #П 16 Г 4 28
20 #P 17 A 1 18
21 #И 9 H 14 23
22 #K 11 M 14 25
23 #A 8 A 1 1
24 #3 8 Г 4 12
25
26
```

Figure 1: Функция для кодирования текста шифром гаммированием конечной

Функция гамма для расшифровки текста. (рис. 2)



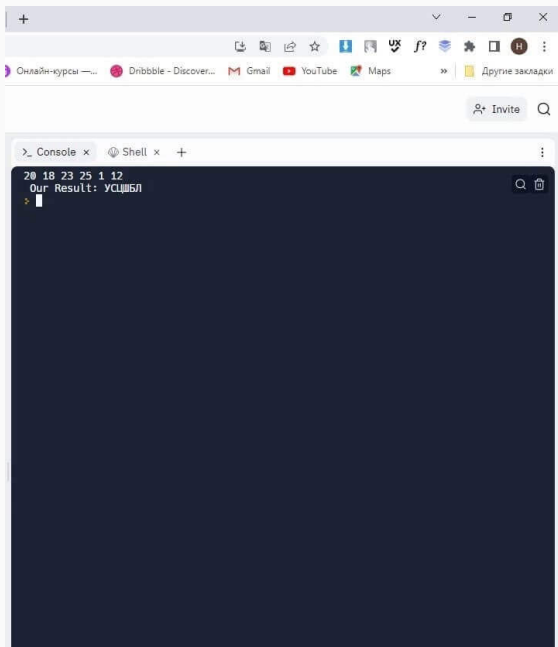
The screenshot shows a web-based IDE with a file explorer on the left and a code editor on the right. The file explorer shows a file named 'main.py'. The code editor displays the following Python code:

```
1
2 alphabet = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
3
4 def gamma(start, key):
5     key+=len(start)
6
7     end_text=""
8     for i, j in enumerate(start):
9         position = alphabet.index(j)
10
11         position_key=alphabet.index(key[i])+1
12         new_position =(position-position_key)%33
13         end_text+=alphabet[new_position]
14         print(new_position, end=" ")
15     print(f"\n Our Result: {end_text}")
16
17 gamma("УСЛББЛ".upper(), "ГAMMA")
18
19 #11 16 17 4 28
20 #P 17 A 1 18
21 #W 9 H 14 23
22 #K 11 M 14 25
23 #A 8 A 1 1
24 #3 8 17 4 12
25
26
```

Figure 2: Функция для расшифровки текста шифром гаммированием конечной

Результат

Результат



The screenshot shows a web browser window with a terminal application running. The browser's address bar is empty, and the page content is a dark-themed terminal window. The terminal has two tabs: ">_ Console x" and "Shell x". The output in the terminal is as follows:

```
20 18 23 25 1 12
Our Result: УСПЕШНО
>
```

The terminal window also features a search icon and a trash icon in the top right corner.

Результат

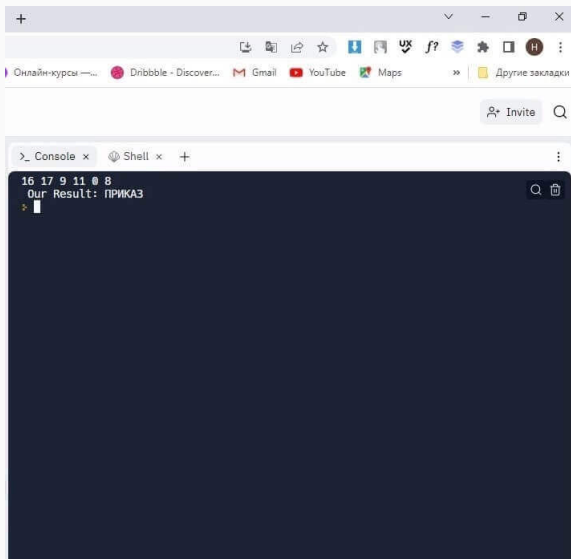


Figure 4: Получение расшифровки текста методом гаммированием конечной гаммой

Реализовал алгоритм шифрования гаммированием конечной гаммой.

Спасибо за внимание