1. Задание

Дан текст на английском языке. Необходимо закодировать его с помощью азбуки Морзе, где каждой букве соответствует последовательность точек и тире. Например, буква «g» превратится в строку «--.». В переменной morze для удобства хранится словарь соответствия латинских букв коду Морзе.

morze = {'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': '--..'}

**На входе:** В одной строке вам дан текст, который состоит из латинских букв и пробелов.

**На выходе:**

Выведите каждое слово исходного текста, закодированное азбукой Морзе. Количество строк в ответе должно совпадать с количеством слов в исходном тексте. Между закодированными буквами ставится ровно один пробел. Например, слово «Help» превратится в «.... . .-.. .--.». Строчные и заглавные буквы кодируются одинаково.

Например:

Ignition sequence start

Перевод

.. --. -. .. - .. --- -.

… . --.- ..- . -. -.-. .

… - .- .-. –

Приложение 1 - Код программы

morze = {'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': '--..'}

def find\_morze(morze):

print("Please input text: ")

text = input()

text = text.lower()

morze\_text = ""

for letter in text:

if letter != " ":

morze\_text += morze[letter.lower()]

else:

morze\_text += '\n'

print(morze\_text)

return

def main():

find\_morze(morze)

return 0

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Задание

В некотором городе открывается новая служба по доставке электронных писем. Необходимо наладить систему регистрации новых пользователей.

Регистрация должна работать следующим образом: если новый пользователь хочет зарегистрироваться на сайте, то он должен послать системе запрос name со своим именем. Система должна определить, существует ли уже такое имя в базе данных. Если такого имени не существует, то оно заносится в базу данных системы и пользователю возвращается ответ "ОК", подтверждающий успешную регистрацию. А если пользователь с таким именем уже существует, то система должна сформировать новое имя и выдать его пользователю в качестве подсказки, при этом сама подсказка также добавляется в базу данных. Новое имя формируется следующим образом: к name последовательно приписываются числа, начиная с 1 (name1, name2 и так далее), и среди них находят такое наименьшее i, что namei еще не содержится в системе.

**Входные данные**

В первой строке входных данных задано число n (1 ≤ n ≤ 100000). Следующие n строк содержат запросы к системе. Каждый запрос представляет собой непустую строку длиной не более 32 символов, состоящую только из строчных букв латинского алфавита.

**Выходные данные**

В выходных данных должно содержаться n строк – ответы системы на запросы: "OK" в случае успешной регистрации, или подсказка с новым именем, если запрашиваемое уже занято.

Приложение 2 – Код программы

def auth():

n = ("")

n = input(n)

if 1 <= int(n) <= 1000000:

names = {}

output = []

for i in range(int(n)):

string = ""

string = input(string)

if string in names:

names[string] += 1

string = string + str(names[string])

names[string] = 0

output.append(string)

else:

names[string] = 0

output.append("OK")

for name in output:

print(name)

return

def main():

auth()

return 0

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Задание

Необходимо создать программу обработки запросов пользователей к файловой системе компьютера. Над каждым файлом можно производить следующие действия: запись – w ("write"), чтение – r ("read"), запуск – x ("execute").

**Входные данные**

На вход программе подаются следующие параметры: число n – количество файлов в файловой системе. В следующих n строках содержится информация с именами файлов и допустимыми действиями (w, x, r), разделенных пробелами. Далее идет число m – количество запросов к файлам вида «операция файл» (обозначение операции: "write", "read", "execute").

**Выходные данные**

Для каждого допустимого запроса программа должна возвращать OK, для недопустимого – Access denied.

Приложение 3 – Код программы

def file\_sys():

n = ""

n = input(n)

names = {}

ans = []

for i in range(int(n)):

file\_name = ""

file\_name = input(file\_name)

file\_name = file\_name.split()

names[file\_name[0]] = file\_name[1:]

m = ""

m = input(m)

for i in range(int(m)):

line = ""

line = input(line)

command = line.split()[0]

file\_name = line.split()[1]

match command:

case "read":

command = "r"

case "write":

command = "w"

case "execute":

command = "x"

case \_:

print("wrong command!")

return

if command in names[file\_name]:

ans.append("OK")

else:

ans.append("Access denied")

for a in ans:

print(a)

return

def main():

file\_sys()

return 0

if \_\_name\_\_ == "\_\_main\_\_":

main()