Практическая работа №1

Выполнил студент группы БВТ2003 Глазков Даниил

Задача 1

Написать функцию, на вход которой подается строка, состоящая из латинских букв. Функция должна вернуть количество гласных букв (a, e, i, o, u) в этой строке.

In [120…
```
string = "asdadadie"
vowels = ['a','e','i','o','u']
total = 0
for s in string:
   if s in vowels:
      total += 1
print (total)
```

```
5
```

Задача 2

Написать функцию на вход, которой подается строка. Функция должна вернуть true, если каждый символ в строке встречается только 1 раз, иначе должна вернуть false.

In [121…
```
arr ="asdqwet13"
setarr = set(arr)
if len(arr) == len(setarr):
    print("true")
else:
    print("false")
```

```
true
```

Задача 3

Написать функцию, которая принимает положительное число и возвращает количество бит равных 1 в этом числе.

In [122…
```
number = 55
bits_count = 0
while number != 0:
    bits_count += number % 2 == 1
    number //= 2
print (bits_count)
```

```
5
```

Задача 4

Написать функцию, которая принимает положительное. Функция должна вернуть то, сколько раз необходимо перемножать цифры числа или результат перемножения, чтобы получилось число, состоящее из одной цифры.
Например, для входного числа:
· 39 функция должна вернуть 3, так как *39=27 => 27=14 => 14=4*
· *4 функция должна вернуть 0, так как число уже состоит из одной цифры*
· *999 функция должна вернуть 4, так как 999=729 => 729=126 => 126=12 => 12=2*

In [123…
```
def digits(n):
    digs =[]
    while n != 0:
        digs.append(n % 10)
        n = n // 10
    return digs

def persistence(n):
    count = 0
    while n > 9:
        digs = digits(n)
        mul = 1
        for digit in digs:
            mul = mul * digit
        n = mul
        print('*'.join(map(str, digs)) + '=' + str(mul))
        count+=1
    return count
print (persistence(999))
```

```
9*9*9=729
9*2*7=126
6*2*1=12
2*1=2
4
```

Задача 5

Написать функция, которая принимает два целочисленных вектора одинаковой длины и возвращает среднеквадратическое отклонение двух векторов.

In [124…
```
a = [11,20,19,17,10]
b = [12,18,19.5,18,9]
c = 0
length = len(a)
for i in range(0, length):
    c += (a[i] - b[i]) ** 2
print (c / length)
```

```
1.45
```

Задача 6

Написать функцию, которая принимает список чисел и возвращает кортеж из двух

элементов. Первый элемент кортежа – мат. ожидание, второй элемент – СКО. Запрещается использовать функции для расчета соответствующих характеристик.

In [4]:
```
import numpy as np
data = [1, 4, 2.5, 3, 7.4, 8]

b = (np.sum(data)/len(data))
D = np.sum([(x - b) ** 2 for x in data]) / len(data)
S = D ** 0.5
print(b, S)
```

```
4.316666666666666 2.556310275029661
```

Задача 7

Написать функцию, принимающая целое положительное число. Функция должна вернуть строку вида "(n1**p1)(n2**p2)...(nk**pk)" **представляющая разложение числа на простые множители (если pi == 1, то выводить только ni).**
Например, для числа 86240 функция должна вернуть "(25)(5)(7**2)(11)"**

In [126…
```
from math import sqrt

def dec_string(number):
    string = ""
    div = 2
    while div ** 2 <= number:
        div_count = 0
        while number % div == 0:
            div_count += 1
            number //= div

        if div_count == 1:
            string += "(" + str(div) + ")"
        elif div_count != 0:
            string += "(" + str(div) + "**" + str(div_count) + ")"

        div += 1

    if number != 1:
        string += "(" + str(number) + ")"

    return string

print(dec_string(86240))
```

```
(2**5)(5)(7**2)(11)
```

Задача 8

Написать функцию, принимающая 2 строки вида "xxx.xxx.xxx.xxx" представляющие ip-адрес и маску сети. Функция должна вернуть 2 строки: адрес сети и широковещательный адрес.м

In [127…
```
import ipaddress
```

```
IP = '192.168.32.16'
MASK = '255.255.0.0'

host = ipaddress.IPv4Address(IP)
net = ipaddress.IPv4Network(IP + '/' + MASK, False)
print('IP:', IP)
print('Mask:', MASK)
print('Subnet:', ipaddress.IPv4Address(int(host) & int(net.netmask)))
print('Broadcast:', net.broadcast_address)
```

```
IP: 192.168.32.16
Mask: 255.255.0.0
Subnet: 192.168.0.0
Broadcast: 192.168.255.255
```

Задача 9

Написать функцию, принимающая целое число n, задающее количество кубиков. Функция должна определить, можно ли из данного кол-ва кубиков построить пирамиду, то есть можно ли представить число n как 1^2+2^2+3^2+…+k^2. Если можно, то функция должна вернуть k, иначе строку “It is impossible”.

In [128…

```
STRING_ANSWER = "It's impossible"

def pyramidka(number):
    progression_num = 1
    sum = 0
    while sum < number:
        sum += progression_num ** 2
        if sum == number:
            return progression_num
        progression_num += 1
    return STRING_ANSWER

print(pyramidka(11))
```

```
It's impossible
```

Задача 10

Написать функцию, которая принимает положительное целое число n и определяющая является ли число n сбалансированным. Число является сбалансированным, если сумма цифр до средних цифр равна сумме цифр после средней цифры. Если число нечетное, то средняя цифра одна, если четное, то средних цифр две. При расчете, средние числа не участвуют.
Например:
· Число 23441 сбалансированное, так как 2+3=4+1
· Число 7 сбалансированное, так как 0=0
· Число 1231 сбалансированное, так как 1=1
· Число 123456 несбалансированное, так как 1+2!=5+6

In [129…

```
def is_balanced(number):
    digits_array = get_digits_array(number)
```

```
    digits_count = len(digits_array)
    middle_position = digits_count // 2
    if digits_count <= 2:
        return False
    elif digits_count % 2 == 0:
        return np.sum(digits_array[:middle_position - 1]) == np.sum(digits_array
    return sum(digits_array[:middle_position]) == sum(digits_array[middle_positi

def get_digits_array(number):
    array = []
    while number > 0:
        array.append(number % 10)
        number //= 10
    return array

print(is_balanced(197628))
```

```
True
```

Задача 11

Написать функцию, которая принимает двумерный массив M в первой колонке которой стоит буква латинского алфавита (данная буква обозначает принадлежность к классу) и вещественное число r, которое удовлетворяет условию 0<r<1. Функция должна разбить данный массив на 2 так, что количество строк в новых массивах пропорционально r и (r-1). Также, в каждом новом массиве, количество строк одного класса должно быть пропорционально количеству строк этого класса в исходном массиве.

In [130…

```
def split_array(arr, r):

    class_dict = {}
    for row in arr:
        class_letter = row[0]
        if class_letter not in class_dict:
            class_dict[class_letter] = []
        class_dict[class_letter].append(row)


    result_array1 = []
    result_array2 = []

    for class_letter, rows in class_dict.items():

        num_rows_1 = int(len(rows) * r)
        num_rows_2 = len(rows) - num_rows_1


        result_array1.extend(rows[:num_rows_1])
        result_array2.extend(rows[num_rows_1:])

    return result_array1, result_array2

M = [['A', 1], ['A', 2], ['A', 3], ['A', 4], ['B', 5], ['B', 6], ['B', 7], ['B',
r = 0.5
result1, result2 = split_array(M, r)
```

```
print("Первый массив:")
print(result1)
print("Второй массив:")
print(result2)
```

```
Первый массив:
[['A', 1], ['A', 2], ['B', 5], ['B', 6], ['C', 9], ['C', 10]]
Второй массив:
[['A', 3], ['A', 4], ['B', 7], ['B', 8], ['C', 11], ['C', 12]]
```