**Инструкция if-elif-else**

Условие:

Из передачи “Здоровье” Аня узнала, что рекомендуется спать хотя бы AA часов в сутки, но пересыпать тоже вредно и не стоит спать более BB часов. Сейчас Аня спит HH часов в сутки. Если режим сна Ани удовлетворяет рекомендациям передачи “Здоровье”, выведите “Это нормально”. Если Аня спит менее AA часов, выведите “Недосып”, если же более BB часов, то выведите “Пересып”.Получаемое число AA всегда меньше либо равно BB.На вход программе в три строки подаются переменные в следующем порядке: AA, BB, HH.

Решение:

A = int(input())

B = int(input())

H = int(input())

A <= B

if A <= H <=B:

print('Это нормально')

elif H < A:

print('Недосып')

elif H > B:

print('Пересып')

**Цикл while**

Условие:

Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

Решение:

c = int (input())

s = 0

while c != 0:

s = s + c

c = int (input())

print (s)

**Цикл for**

Условие:

Напишите программу, которая считывает с клавиатуры два числа aa и bb, считает и выводит на консоль среднее арифметическое всех чисел из отрезка [a; b][a;b], которые кратны числу 33.

Решение:

a = int(input())

b = int(input())

s = 0

c = 0

for j in range (a,b+1):

if j%3 == 0:

s = s+j #42

c = c+1

j+=1

print(s/c)

**Операторы break, continue**

Условие:

Напишите программу, которая считывает целые числа с консоли по одному числу в строке.

Для каждого введённого числа проверить:

если число меньше 10, то пропускаем это число;

если число больше 100, то прекращаем считывать числа;

в остальных случаях вывести это число обратно на консоль в отдельной строке.

Решение:

a = 0

while a<=100:

a = int (input())

if a > 100:

break

if a<10:

continue

print(a)

**Списки**

Условие:

Напишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его соседей. Для элементов списка, являющихся крайними, одним из соседей считается элемент, находящий на противоположном конце этого списка. Например, если на вход подаётся список "1 3 5 6 10", то на выход ожидается список "13 6 9 15 7" (без кавычек).

Решение:

s = [ int(i) for i in input().split()]

t = []

l = len(s)-1

k = 0

i = 0

if len(s)==0:

print(str(0))

else:

for st in s:

if len(s)>1:

if i==0:

k = s[i+1] + s[-1]

t.append(k)

elif i>0 and i<l:

k=s[i-1]+s[i+1]

t.append(k)

elif i==l:

k = s[i-1]+s[0]

t.append(k)

elif len(s)==1:

k = s[i]

t.append(k)

i +=1

j = 0

for st2 in t:

print(str(t[j])+' ',end='')

j +=1

**Индексы**

Условие:

Кодирование осуществляется следующим образом:

s = 'aaaabbсaa' преобразуется в 'a4b2с1a2', то есть группы одинаковых символов исходной строки заменяются на этот символ и количество его повторений в этой позиции строки.

Напишите программу, которая считывает строку, кодирует её предложенным алгоритмом и выводит закодированную последовательность на стандартный вывод.

Решение:

s = str(input())

l = len(s)-1

c = 1

t = ''

if len(s)==1:

t = t +s+str(c)

else:

for i in range(0,l):

if s[i]==s[i+1]:

c +=1

elif s[i]!=s[i+1]:

t = t + s[i]+str(c)

c = 1

for j in range(l,l+1):

if s[-1]==s[-2]:

t = t +s[j]+str(c)

elif s[-1]!=s[-2]:

t = t +s[j]+str(c)

c = 1

print(t)

**Словари (dict) и работа с ними**

Условие:

Программа должна считывать одну строку со стандартного ввода и выводить для каждого уникального слова в этой строке число его повторений. Каждое уникальное слово﻿ должно выводиться только один раз.

Решение:

n=''

n = str(input())

m = []

m.append([str(s.lower()) for s in n.split()])

d = {}

li, lj = len(m), len(m[0])

for i in range(li):

for j in range(lj):

p = m[i][j]

if p in d:

d[p]+=1

else:

d[p] = 1

for key,value in d.items():

print(key,value)

**Множества**

Условие:

Для каждого города из множества напечатайте на экран сообщение «У меня есть друг в городе <название\_города>.»

Решение:

cities = [

'Вологда',

'Чебоксары',

'Тольятти',

'Москва',

'Бремен',

'Санкт-Петербург',

'Новороссийск',

'Челябинск',

'Вологда',

'Новосибирск',

'Челябинск',

'Санкт-Петербург',

'Москва',

'Новосибирск'

]

unique\_cities = set(cities)

for i in unique\_cities:

print('У меня есть друг в городе ' + i)

**Функции и их аргументы**

Условие:

Напишите функцию, которая принимает на вход список целых чисел, удаляет из него все нечётные значения, а чётные нацело делит на два.

Решение:

def modify\_list(l):

le = len(l)-1

i = le

while i!=-1:

if l[i]%2:

del l[i]

else:

l[i]=l[i]//2

i -=1

return

Условие:

Сделайте так, чтобы при вызове функции без аргументов она напечатала фразу «Друг, пойдём учить Python.»

Решение:

def lets\_go(name='Друг', target='учить Python'):

print(name + ', пойдём ' + target)

name='Друг'

target='учить Python'

lets\_go('Гарри', 'ловить Волан-де-Морта')

lets\_go()

**Конструкция try - except для обработки исключений**

Условие:

Напишите функцию:

Выполните HTTP-запрос, поместив вызов функции get() внутрь блока try.

Значения URL и параметров получите из функций make\_url() (в неё нужно передать нужный город как аргумент city) и make\_parameters().

При выбрасывании исключения типа requests. ConnectionError — функция what\_weather() должна возвращать сообщение об ошибке '<сетевая ошибка>'.

Если код HTTP-ответа равен 200, верните из функции текст ответа. В противном случае функция должна вернуть строку '<ошибка на сервере погоды>'.

Решение:

import requests

cities = [

'Омск',

'Калининград',

'Челябинск',

'Владивосток',

'Красноярск',

'Москва',

'Екатеринбург'

]

def make\_url(city):

return f'http://wttr.in/{city}'

def make\_parameters():

params = {

'format': 2, # погода одной строкой

'M': '' # скорость ветра в "м/с"

}

return params

def what\_weather(city):

try:

request = requests.get(make\_url(city), params=make\_parameters())

if request.status\_code == 200: return request.text

else: return '<ошибка на сервере погоды>'

except requests.ConnectionError : return '<сетевая ошибка>'

print('Погода в городах:')

for city in cities:

print(city, what\_weather(city))

**Файлы**.

Условие:

digits = set('0123456789')

i = 0

multiplier = ''

decrypted = ''

with open('03\_04\_02\_input.txt') as in\_f\_obj:

string = in\_f\_obj.readline().strip()

char = string[i]

i += 1

while i < len(string):

while string[i] in digits:

multiplier += string[i]

i += 1

if i > (len(string) - 1):

break

decrypted += (char \* int(multiplier))

multiplier = ''

if i > (len(string) - 1):

break

char = string[i]

i += 1

with open('03\_04\_02\_ouput.txt', 'w') as out\_f\_obj:

out\_f\_obj.write(decrypted)