Тверской Государственный технический университет

Кафедра: Программного обеспечения и вычислительной

техники

Лабораторная работа № 1

7 10

Выполнил: Студент второго курса

Группы Б.ПИН.РИС 18.06

Федотов Ярослав Всеволодович

Задача №1

Вычислите в python значение выражения:

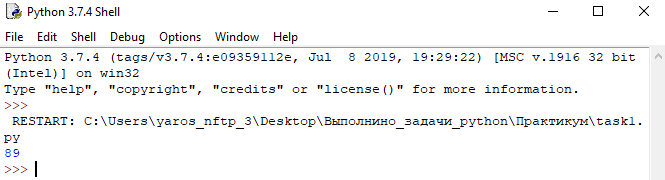
9\*\*19 - int(float(9\*\*19))

c.

a=9\*\*19-int(float(9\*\*19))

print(a)

d.



Задача №2

a.

Напишите программу. Тимофей обычно спит ночью X часов и устраивает себе днем тихий час на Y минут. Определите, сколько всего минут Тимофей спит в сутки.

c.

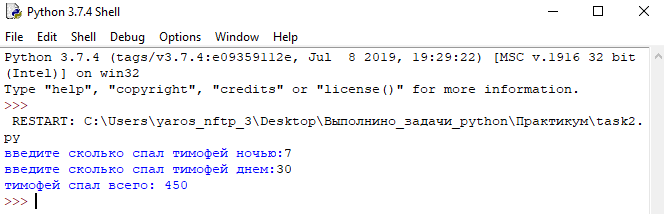
x = int(input("введите сколько спал тимофей ночью:"))\*60

y = int(input("введите сколько спал тимофей днем:"))

s = (int(x+y))

print("тимофей спал всего:", s, "минут")

d.



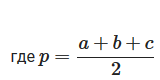
Задача №3

a.

В то далёкое время, когда Паша ходил в школу, ему очень не нравилась формула Герона для вычисления площади треугольника, так как казалась слишком сложной. В один прекрасный момент Павел решил избавить всех школьников от страданий и написать и распространить по школам программу, вычисляющую площадь треугольника по трём сторонам.

Одна проблема: так как эта формула не нравилась Павлу, он её не запомнил. Помогите ему завершить доброе дело и напишите программу, вычисляющую площадь треугольника по переданным длинам трёх его сторон по формуле Герона:

**

– полупериметр треугольника. На вход программе подаются целые числа, выводом программы должно являться вещественное число, соответствующее площади треугольника.

c.

a=int(input("введите первую сторону треугольника:"))

b=int(input("введите вторую сторону треугольника:"))

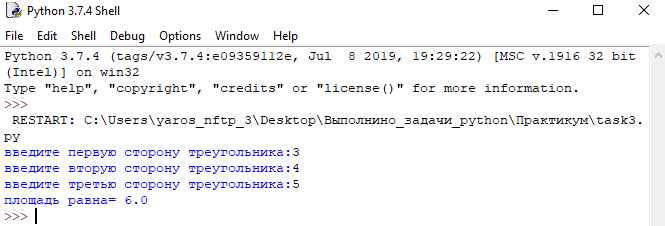
c=int(input("введите третью сторону треугольника:"))

p=(a+b+c)/2

s=(p\*(p-a)\*(p-b)\*(p-c))\*\*0.5

print("площадь равна=",s)

d.



Задача №4

a.

Напишите программу, принимающую на вход целое число, которая выводит True, если переданное значение попадает в интервал (−15,12]∪(14,17)∪[19,+∞)

и False в противном случае (регистр символов имеет значение).

c.

k=int(input("введите число для интервала:"))

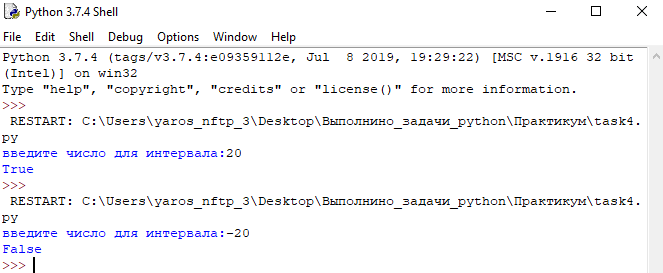
if (-15<k<13) or (14<k<17) or (k>=19):

print(True)

else:

print(False)

d.



Задача №5

a.

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, \*, mod, pow, div, где   
mod — это взятие остатка от деления,   
pow — возведение в степень,   
div — целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!". Обратите внимание, что на вход программе приходят вещественные числа.

c.

a=float(input("введите первое число:"))

b=float(input("введите второе число:"))

c=input("введите тип операции:")

if c=="+":

print (a+b)

elif c=="-":

print (a-b)

elif c=="\*":

print (a\*b)

elif c=="/":

if b==0:

print("деление на 0!")

else:

print (a/b)

elif c=="pow":

print (a \*\* b)

elif c=="mod":

if b==0:

print("деление на 0!")

else:

print (a % b)

elif c=="div":

if b==0:

print("деление на 0!")

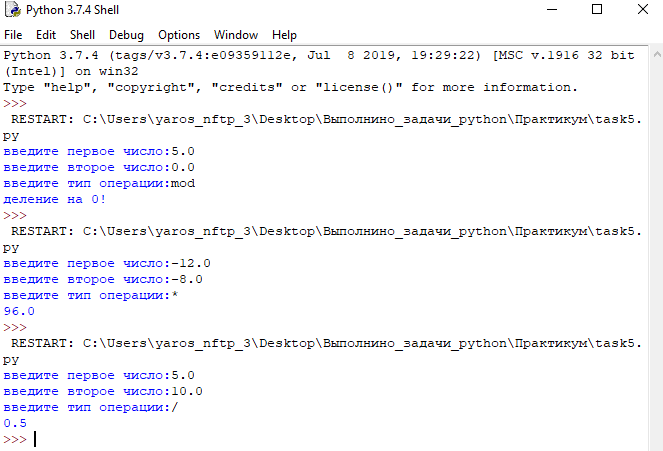
else:

print (a // b)

else:

print("введена несушествующая операция.")

d.



Задача №6

a.

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

c.

temp = 0

a = int(input())

b = int(input())

if b > a:

temp = a

a = b

b = temp

c = int(input())

if c > a:

temp = a

a = c

c = temp

elif c < b:

temp = b

b = c

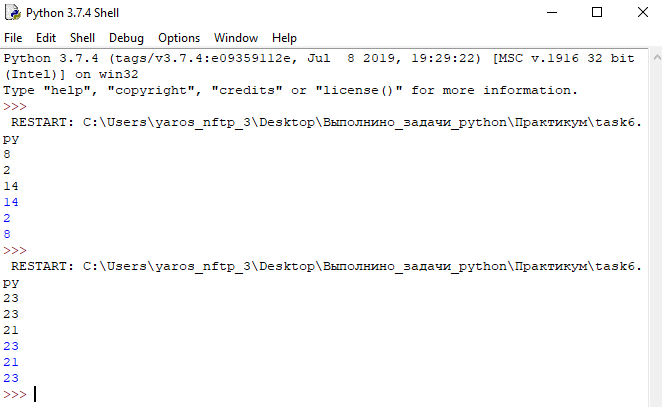
c = temp

print (a)

print (b)

print (c)

d.



Задача №7

a.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.  
  
Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.  
  
На вход программе подаётся строка из шести цифр.

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

c.

namber = str(input())

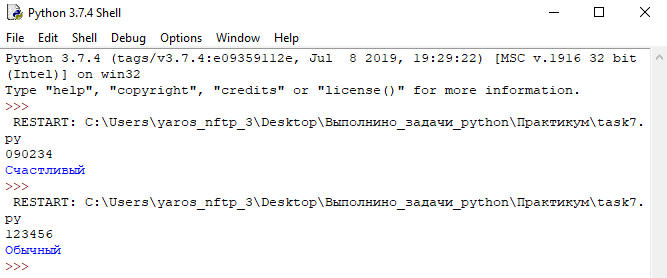
if int(namber[0]) + int(namber[1]) + int(namber[2]) == int(namber[3]) + int(namber[4]) + int(namber[5]):

print("Счастливый")

else:

print("Обычный")

d.



Задача №8

a.

В институте биоинформатики по офису передвигается робот. Недавно студенты из группы программистов написали для него программу, по которой робот, когда заходит в комнату, считает количество программистов в ней и произносит его вслух: "n программистов".

Для того, чтобы это звучало правильно, для каждого *n*

нужно использовать верное окончание слова.

Напишите программу, считывающую с пользовательского ввода целое число *n*

(неотрицательное), выводящее это число в консоль вместе с правильным образом изменённым словом "программист", для того, чтобы робот мог нормально общаться с людьми, например: 1 программист, 2 программиста, 5 программистов.

В комнате может быть очень много программистов. Проверьте, что ваша программа правильно обработает все случаи, как минимум до 1000 человек.

c.

n = str(input())

e = int(n[len(n) - 1])

if e == 1:

print(n, "Програмист")

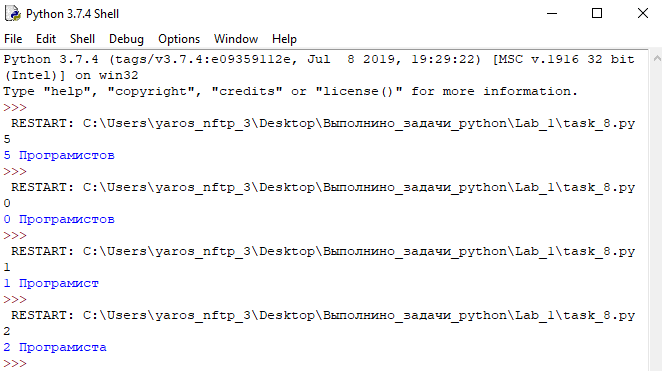
elif 0 < e < 5:

print(n, "Програмиста")

else:

print(n, "Програмистов")

d.



Задача №9

a.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и сразу после этого выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, после этого считывание продолжать не нужно.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.

c.

step = int(input())

sum = step

stepen = step \* step

sumstepen = stepen

while sum != 0:

step = int(input())

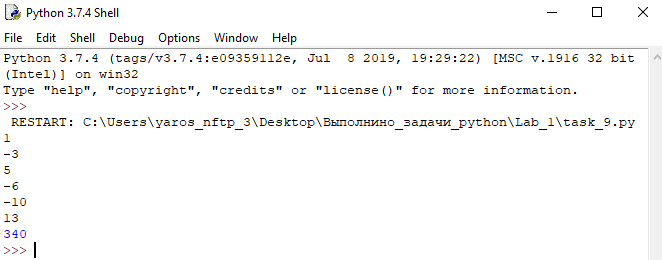
sum += step

stepen = step \* step

sumstepen += stepen

print(sumstepen)

d.



Задача №10

a.

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если n = 7, то программа должна вывести 1 2 2 3 3 3 4.

c.

r = int(input())

step = int(1)

string = str("")

for x in range(r - 1):

for i in range(step):

if len(string) == r:

print(string)

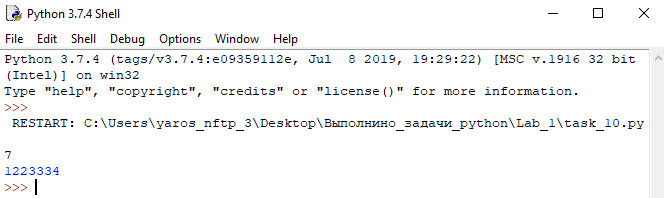
raise SystemExit()

string += str(step)

step += 1

print(string)

d.



Задача №11

a.

Напишите программу, которая считывает список чисел lst из первой строки и число x из второй строки, которая выводит все позиции, на которых встречается число x в переданном списке lst.

Позиции нумеруются с нуля, если число x не встречается в списке, вывести строку "Отсутствует" (без кавычек, с большой буквы).

Позиции должны быть выведены в одну строку, по возрастанию абсолютного значения.

c.

massive = str(input())

n = int(input())

string = ""

for i in range(len(massive)):

if int(massive[i]) == n:

string += str(i) + " "

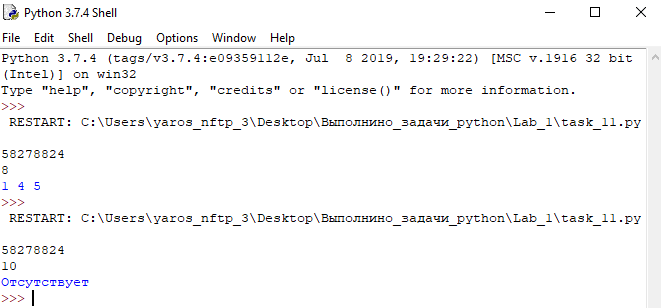
if string == "":

print("Отсутствует")

else:

print(string)

d.



Задача №12

a.

Напишите программу, на вход которой подаётся прямоугольная матрица в виде последовательности строк, заканчивающихся строкой, содержащей только строку "end" (без кавычек)

Программа должна вывести матрицу того же размера, у которой каждый элемент в позиции i, j равен сумме элементов первой матрицы на позициях (i-1, j), (i+1, j), (i, j-1), (i, j+1). У крайних символов соседний элемент находится с противоположной стороны матрицы.

В случае одной строки/столбца элемент сам себе является соседом по соответствующему направлению.

b.

1) Введённая матрица дополняется рамкой из крайних элементов.

2) Создаётся матрица каждый элемент которой равен сумме соседей элемента, на той же позиции, в матрице, полученной в первом шаге.

c.

baf = ""

p = ""

tempstr = ""

mas = []

masres = []

masdop = []

while baf != "end":

baf = str(input())

if baf != "end":

mas.append([int(j) for j in baf.split(' ')])

for f in range(len(mas)):

if f == 0:

for h in range (len(mas[f])):

p += str(mas[0][h]) + ' '

s = str(mas[0][len(mas[f]) - 1]) + ' ' + str(p) + str(mas[0][0])

p = ""

masdop.append([int(j) for j in s.split(' ')])

for l in range (len(mas[f])):

p += str(mas[len(mas) - 1 - f][l]) + ' '

s = str(mas[len(mas) - 1 - f][len(mas[f]) - 1]) + ' ' + str(p) + str(mas[len(mas) - 1 - f][0])

p = ""

masdop.append([int(j) for j in s.split(' ')])

if f == len(mas) - 1:

for a in range (len(mas[f])):

p += str(mas[len(mas) - 1 - f][a]) + ' '

s = str(mas[len(mas) - 1 - f][len(mas[f]) - 1]) + ' ' + str(p) + str(mas[len(mas) - 1 - f][0])

p = ""

masdop.append([int(j) for j in s.split(' ')])

for x in range (len(mas[f])):

p += str(mas[f][x]) + ' '

s = str(mas[f][len(mas[f]) - 1]) + ' ' + str(p) + str(mas[f][0])

p = ""

masdop.append([int(j) for j in s.split(' ')])

if 0 < f < len(mas) - 1:

for v in range (len(mas[f])):

p += str(mas[len(mas) - 1 - f][v]) + ' '

s = str(mas[len(mas) - 1 - f][len(mas[f]) - 1]) + ' ' + str(p) + str(mas[len(mas) - 1 - f][0])

p = ""

masdop.append([int(j) for j in s.split(' ')])

for i in range(1, len(mas) + 1):

for j in range(1, len(mas[0]) + 1):

tempstr += str(masdop[i - 1][j] + masdop[i + 1][j] + masdop[i][j - 1] + masdop[i][j + 1])

tempstr += " "

masres.append(tempstr)

tempstr = ""

masres.reverse()

for e in range(0, len(masres)):

print(masres[e])

d.

