

Разные задачи для решения на Python темам)

Условный оператор

Задача 6. Проверка логина и пароля

Даны два имени пользователя (bender, calculon) и пароли этих пользователей (BENDER и CALCULON).

Необходимо написать программу, принимающую от пользователя логин и выводит одно из следующих сообщений:

- "successful connection", если проверка прошла успешно;
- "unsuccessful connection", если проверка не прошла;
- "unknown user", если введено неизвестное программе имя пользователя

B [1]:

```
1 bender_login = 'bender'
2 bender_password = 'BENDER'
3 calculon_login = 'calculon'
4 calculon_password = 'CALCULON'
5 login = input('login: ')
6 password = input('password: ')
7
8 if login == bender_login:
9     if password == bender_password:
10         print('successful connection')
11     else:
12         print('unsuccessful connection')
13 elif login == calculon_login:
14     if password == calculon_password:
15         print('successful connection')
16     else:
17         print('unsuccessful connection')
18 else:
19     print("unknown user")
```

```
login: benber
password: RUDN IS THE BEST UNIVERSITY IN RUSSIA
unknown user
```

Тесты:

Входные данные:

bender

BENDER

Выходные данные: successful connection

Входные данные:

calculon

BENDER

Выходные данные: unsuccessful connection

Входные данные:

crushinator

Выходные данные: unknown user

Входные данные:

user1

Выходные данные: unknown user

Входные данные: calculon

CALCULON

Функции. Начало

Задача 5. Оценка за курс

Университет вычисляет итоговые оценки студентов на основе их промежуточных экзаменационных баллов. Итоговая оценка определяется путем взятия 40 и 60% от оценки за итоговый экзамен. Если итоговая оценка больше 60 баллов, то студент успешно пройденным.

Напишите функцию Python, которая принимает на вход промежуточные и итоговые оценки и возвращает строку "Зачет" в случае успешного прохождения курса и "Незачет" в случае неуспешного прохождения курса.

Подсказка. Используйте только операторы if и основные арифметические операторы, а также определите свои собственные функции.

In [2]:

```
1 term_score = int(input('Enter term score: '))
2 exam_score = int(input('Enter exam score: '))
3
4 score = term_score*0.4 + exam_score*0.6
5
6 if score > 60:
7     print("Зачет")
8 else:
9     print("Незачет")
```

Enter term score: 57

Enter exam score: 90

Зачет

Тесты:

Входные данные:

50

70

Выходные данные: Зачет

Входные данные:

30

50

Выходные данные: Незачет

Входные данные: 40

60

Выходные данные: Незачет

Входные данные:

80

90

Выходные данные: Зачет

Входные данные:

70

40

Выходные данные: Незачет

Строки

Задача 3. Проверка паспорта

Известно, что номера паспортов в России состоят из 6 цифр. Допустим, что начинаются с номеров 75 и 76.

Напишите функцию, принимающую на вход номер паспорта и возвращающую номер паспорта нового образца, и "old", если передан номер паспорта старого образца. Если функция должна вернуть "wrong number", если передана строка, не соответствующая формату номера паспорта.

Выведите на экран результат работы функции.

In [6]:

```
1 number = input('Enter number of passport: ')
2 n = str(number[:2])
3
4 if number.isnumeric() == False:
5     print('wrong number')
6 elif len(str(number)) != 6:
7     print('wrong number')
8 elif (n == '75' or n == '76'):
9     print('this passport is new')
10 else:
11     print('this passport is old')
```

Enter number of passport: 764756
this passport is new

Тесты:

Входные данные:

751234

Выходные данные: new

Входные данные:

551234

Выходные данные:old

Входные данные:

123asd

Выходные данные: wrong number

Входные данные:

7654321

Выходные данные: wrong number

Входные данные:

1245

Выходные данные: wrong number

Задача 9. Обращения маркетологов

Менеджер Игорь получает очень много сообщений от назойливых маркетс интересно, сколько раз они обращаются к нему по имени в письмах. Помо количество вхождений его имени в письмо, поданное на вход.

Подсказка. Требуется посчитать именно количество вхождений имени ' всевозможных видоизменений. Но учтите, что имя может быть написа.

В [7]:

```
1 text = input('Enter message: ')
2 t = text.lower()
3 print(t.count("игорь"))
```

Enter message: Игорь. ИГОРЬ! ТЫ ВЫИГРАЛ МИЛЛИ.. читать далее..
2

Тесты:

Входные данные:

некоторое письмо, Игорь

Выходные данные: 1

Входные данные:

вам снова письмо, Игорь. Почему вы не отвечаете, Игорь?

Выходные данные: 2

Входные данные:

спишь?

Выходные данные: 0

Входные данные:

Игорь, узнайте тайну имени Игорь всего за 100 рублей отправив смс с кодом 5553535

Выходные данные: 3

Входные данные:

Задача 10. Подсчёт предложений

В переменной `text` сохранен некоторый текст. Посчитайте количество предложений в тексте и выведите их сумму на экран. Если текст состоит из одного предложения, то вывести "слишком короткий текст".

Подсказка. Количество предложений равно количеству точек.

В [14]:

```
1 text = input('Enter text of message: ')
2
3 if text.count(".") <= 1:
4     print("слишком короткий текст")
5 else:
6     print(text.count("."))
```

Enter text of message: Привет. Меня зовут Олег. И я супер-пупер
3

Тесты:

Входные данные:

Я короткий текст.

Выходные данные: слишком короткий текст

Входные данные:

Я текст подлиннее, но все еще короткий текст.

Выходные данные: слишком короткий текст

Входные данные:

Я очень недлинный текст. Но я еще вырасту. Вот увидите, я вам всем пока

Выходные данные: 3

Входные данные:

Я-я еще расту. Не смотрите на меня. Скоро все будет. Но пока еще не готов

Выходные данные: 4

Циклы

Задача 6. Сумма целых чисел

Дано целое положительное число N. Выведите максимально возможную сумму целых чисел от 1 до n так, чтобы эта сумма была строго меньше заданного

В [10]:

```
1 N = int(input('Enter N: '))
2 a=1
3 b=0
4 while a<=N:
5     b+=a
6     c=b-a
7     if b>=N:
8         print(c)
9         break
10    a+=1
```

Enter N: 7

6

Тесты:

Входные данные: 100 Выходные данные: 91

Входные данные: 1000 Выходные данные: 990

Входные данные: 10 Выходные данные: 6

Входные данные: 3 Выходные данные: 1

Входные данные: 20 Выходные данные: 15

Массивы

Задача 6. Главная диагональ матрицы

Вам дается целое число n. Выведите матрицу размера nxn, в которой все диагонали, на которой стоят 1. (Главная диагональ квадратной матрицы с индексами i и j, в которых i=j. То есть это элементы с индексами 00, 11, 22

B [13]:

```
1 n = int(input('Enter n: '))
2 matrix = []
3
4 for i in range(n):
5     a = []
6     for j in range(n):
7         if i == j:
8             a.append(1)
9         else:
10            a.append(0)
11    matrix.append(a)
12
13 for i in range(n):
14    print(matrix[i])
```

Enter n: 5

```
[1, 0, 0, 0, 0]
[0, 1, 0, 0, 0]
[0, 0, 1, 0, 0]
[0, 0, 0, 1, 0]
[0, 0, 0, 0, 1]
```

Тесты:

Входные данные: 5

Выходные данные:

```
[1, 0, 0, 0, 0]
[0, 1, 0, 0, 0]
[0, 0, 1, 0, 0]
[0, 0, 0, 1, 0]
[0, 0, 0, 0, 1]
```

Входные данные: 4

Выходные данные: [1, 0, 0, 0]

```
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]
```

Входные данные: 1

Выходные данные:

```
[1]
```

Входные данные: 2

Выходные данные:

```
[1, 0]
[0, 1]
```

Словари и множества

Задача 5. Города и страны

В словаре `countries` каждый ключ — это название страны, а его значения — городов, которые в этой стране расположены. Ваша задача написать программу, которая принимает название города и сохраняет ее в переменную `city`, а на выходе выводит, в какой стране находится указанный город. Если страна успешно найдена, то выводит "INFO: <указанный_город> is a city in <найденная_страна>". В противном случае выводит "ERROR: Country for <указанный_город> is not found"

В [15]:

```
1 countries = {
2     'Russia': ['Moscow', 'Saint Petersburg', 'Novosibirsk'],
3     'Germany': ['Berlin', 'Hamburg', 'Munich'],
4     'France': ['Paris', 'Toulouse', 'Marseille'],
5     'England': ['London', 'Birmingham', 'Manchester'],
6     'Spain': ['Madrid', 'Barcelona', 'Valencia']
7 }
8 city = input()
9
10 found = False
11 for country, cities in countries.items():
12     if city in cities:
13         found = True
14         print('INFO:', city, "is a city in", country)
15         break
16
17 if not found:
18     print('ERROR: Country for', city, 'is not found')
```

Berlin

INFO: Berlin is a city in Germany

Тесты:

Входные данные: Birmingham

Выходные данные: INFO: Birmingham is a city in England

Входные данные: Moscow

Выходные данные: ERROR: Country for Moscow is not found

Входные данные: England

Выходные данные: ERROR: Country for England is not found

Входные данные: trondheim

Выходные данные: ERROR: Country for trondheim is not found

Входные данные: Paris

Выходные данные: INFO: Paris is a city in France

Алгоритмы со строками

Задача 1. Проверка на палиндром 1

Проверьте, является ли введенное число палиндромом. Число может быть в виде десятичной дроби.

В [16]:

```
1 num = input('Enter number: ')
2 l = []
3 a = []
4 for i in num:
5     l.append(i)
6     a.append(i)
7
8 for i in l:
9     if i==" ":
10         c=l.index(i)
11         l.pop(c)
12
13 for i in a:
14     if i==" ":
15         d=a.index(i)
16         a.pop(d)
17
18 a.reverse()
19
20 if a==l:
21     print('True')
22 else:
23     print('False')
```

Enter number: 45
False

Тесты:

Входные данные: 101
Выходные данные: True

Входные данные: 12490
Выходные данные: False

Входные данные: 123.321
Выходные данные: True

Входные данные: 1233.21
Выходные данные: True

Входные данные: 123.567
Выходные данные: False

3*

Строки

Задача 11. Акула

Ваш друг пригласил вас на плавучий остров на отдалении в 1 километр от берега. На острове располагается огромная горка для прыжков в воду, к ней ведет извилистая дорожка. Вы взбираетесь наверх, толчок, скольжение, взлет!

Во время полета с горки вы замечаете исчезающий в глубине силуэт акулы.

В рамках задачи вам, по большому счету, нужно выяснить, кто доплывет до острова раньше: вы или акула?

Не забывайте, что акуле нужно преодолеть большее расстояние.

Если получается убежать от акулы, то нужно вывести – "Спасен от акулы!"

Для удобства считаем, что события происходят в одномерном пространстве значений:

sharkDistance – расстояние до акулы;

sharkSpeed – скорость акулы;

pontoonDistance – расстояние, которое нужно проплыть обратно до берега;

yourSpeed – ваша скорость;

dolphin – логическая переменная, если "true", то можно уменьшить скорость, если "false", то дельфин будет её отвлекать.

B [17]:

```
1 sharkDistance = int(input())
2 sharkSpeed = int(input())
3 pontoonDistance = int(input())
4 yourSpeed = int(input())
5 dolphin = input()
6
7 if dolphin == "true":
8     sharkSpeed = sharkSpeed/2
9
10 sharkTime = sharkDistance / sharkSpeed
11 yourTime = pontoonDistance / yourSpeed
12
13 if yourTime <= sharkTime:
14     print("Спасен от акулы!")
15 else:
16     print("Акула съела пловца!")
```

745

45

32

34

32

Спасен от акулы!

Тесты:

Входные данные:

5 1

2 2

true

Выходные данные: Спасен от акулы!

Входные данные:

5 7

2 1

false

Выходные данные: Акула съела пловца!

Входные данные:

6 3

4 1

false

Выходные данные: Акула съела пловца!

Входные данные:

6 3

4 1

true

Выходные данные: Спасен от акулы!

Массивы

Задача 11. Обратная диагональ матрицы

Мы решали задание с заполнением главной диагонали, а теперь нам предстоит для обратной (побочной) диагонали.

Выведите матрицу размера $n \times n$ с такими элементами: все значения, стоящие на обратной диагонали, равны 1, остальные - 0. (Побочная диагональ квадратной матрицы — это диагональ элементов с индексами ij , в которых $i+j=n-1$. То есть в данном случае это э

В [18]:

```
1 matrix = []
2 n = int(input('Enter n: '))
3
4 for i in range(n):
5     matrix.append([0]*n)
6
7 for i in range(n):
8     for j in range(n):
9         if i + j == n - 1:
10            matrix[i][j] = 1
11
12 for i in range(n):
13     print(matrix[i])
```

```
Enter n: 4
[0, 0, 0, 1]
[0, 0, 1, 0]
[0, 1, 0, 0]
[1, 0, 0, 0]
```

Тесты:

Входные данные: 5

Выходные данные:

```
[0, 0, 0, 0, 1]
[0, 0, 0, 1, 0]
[0, 0, 1, 0, 0]
[0, 1, 0, 0, 0]
[1, 0, 0, 0, 0]
```

Входные данные: 1

Выходные данные:

```
[1]
```

Входные данные: 2 Выходные данные:

```
[0, 1]
[1, 0]
```

Входные данные: 3

Выходные данные: [0, 0, 1]

0 1 0