

S1- Bir string girdi alıp, verilen string içinde hangi harfin kaç tane olduğunu bulan bir fonksiyon yazın.

Örneğin girdi “kara murat kim” olsun,

Çıktı: {“k”:2, “a”:3, “r”:2, “m”:2, “u”:1, “i”:1, “ ”:2}

S2- Yukarıdaki gibi bir sözlük alıp bu sözlük içindeki harfleri sayılarına göre birer liste içinde toparlayıp yeni bir sözlük oluşturan bir fonksiyon yazın.

Örneğin: yeni_sozluk1 = {1: ['u', 't', 'i'], 2: [' ', 'r', 'k', 'm'], 3: ['a']}

S3- Siz bir öğretmensiniz ve sınıfınıza 2 sınav yaptınız. Sınavların sonuçları aşağıda bir sözlük olarak verilmektedir. İlk sınav notlarının ortalamasını bulan ve döndüren bir işlev yazın.

```
sinav_sozluk = {"john":{"exam1":40, "exam2":70}, "murat":{"exam1":20, "exam2":30},  
"faruk":{"exam1":35, "exam2":55}, "sow":{"exam1":80, "exam2":90}}
```

S4- S3 için yazdığınız fonksiyonu değiştirin ve tüm ortalamayı hesaplayan bir fonksiyon yazın.

(Toplam ortalama = Tüm notlar sınav1 ve sınav2 / 2x(öğrenci sayısı))

S5- `sinav_sonuc` adlı bir sınav sonuç listeniz var. ([öğrenci_adı, (sınav1_not, sınav2_not)])

```
sinav_sonuc = ["john", (40, 70), "murat", (20, 30), "faruk", (35, 55), "sow", (80, 90)]
```

İlk sınav notlarının ortalamasını bulan ve döndüren bir fonksiyon yazın.

S6- S5 için yazdığınız fonksiyonu değiştirin ve tüm ortalamayı hesaplayan bir fonksiyon yazın.

S7- Aşağıdaki kodu okuyun ve çıktığı söyleyin.

```
def blabla(x):  
    for i,j,k in x:  
        print(i,j,k)  
  
blabla([("batman", "beats", "superman"), ("Sixth", "of", "November"), ("Since", "18",  
"years"), ("You are", "the", "best")])
```

S8- Aynı çıktıyı almak için Q7'deki fonksiyonu değiştirin, ancak bu kez döngü içinde birden çok değişken kullanmayın. (for i, j, k in x yerine, for i in x) **S9-** Bunu:

{'limon': 1, 'karpuz': 5, 'elma': 1, 'muz': 2} Buna

dönüştüren bir fonksiyon yazın:

[('karpuz', 5), ('limon', 1), ('elma', 1), ('muz', 2)]

S10- Bunu:

[('karpuz', 5), ('limon', 1), ('elma', 1), ('muz', 2)] Buna:

{'limon': 1, 'karpuz': 5, 'elma': 1, 'muz': 2} geri

dönüştüren bir fonksiyon yazın.

S11- Bu alıştırmada, peşpeşe aynı karakterlerden oluşan stringleri sahip dizeleri sıkıştıracağız. Programınız önce kullanıcıdan bir string girmesini istemeli ve sıkıştırılmış bir string olarak döndürmelidir. Örneğin, girdi “aaaabbbaabbbbbbcccd” ise, çıktı “4a2b3a5b3c1d” olmalıdır. (Uyarı: Harfleri saymak için bir sözlük kullanın).

S12- İki sözlük alan bir fonksiyon yazın (anahtarlar ve değerlerin tamsayı olduğu s1 ve s2) ve s1 ve s2'deki öğelerin birleşimini içeren yeni bir sözlük döndürür. s1 ve s2 adlı iki sözlüğün birleştiği yeni sözlük s3 olsun:

- Eğer bir anahtar yalnızca s1'de varsa, bu anahtar ve s1'deki karşılık gelen değeri s3'e dahil edilmelidir.
- Eğer bir anahtar yalnızca s2'de varsa, bu anahtar ve s2'deki karşılık gelen değeri s3'e dahil edilmelidir.
- Eğer aynı anahtar hem s1 hem de s2'de varsa, s3 bu anahtarı ve s1 ve s2 içinden karşılık gelen değerlerden büyük olanını içermelidir.

Örnek: s1 = {1: 2, 3: 4, 5: 6} ve s2 = {1: 3, 6: 5, 3: 8} ise çıktınız {1: 3, 5: 6, 6: 5, 3: 8}. gibi bir şey olmalıdır. Çıktı sözlüğündeki öğelerin sırası önemli değildir.

S13- Bir kategori (“havaalanı”, “şehir” veya “durum” dan biri olabilir) ve bir sözlük (anahtarları :havaalanı adı, şehir adı, ve uçuş türünden ("yurtiçi" veya "uluslararası") oluşan bir tuple ve değerleri saat başı kalkan uçak sayısı olan) alan genelleştir() isimli bir fonksiyon oluşturun. Fonksiyon anahtarları kategorilerden ve değerleri toplam uçuş sayılarından oluşan bir sözlük döndürmelidir

Örnek: Girdi ornek_sozluk (aşağıda tanımlanmıştır) ve “şehir” ise, çıktı {“İstanbul”: 135000, İzmir: 20000} gibi bir şey olmalıdır.

ornek_sozluk = {(“Atatürk”, “İstanbul”, ”Uluslararası”): 30000, (“Atatürk”, “İstanbul”, ”Yurtiçi”): 65000, “SabihaGokcen”, “İstanbul”, ”Uluslararası”): 10000, “SabihaGokcen”, “İstanbul”, ”Yurt İçi”): 30000, (“Menderes”, “İzmir”, ”Uluslararası”): 5000, (“Menderes”, “İzmir”, ”Yurt İçi”): 15000}

S14- Bir stringi parametre olarak alan charMap adlı bir fonksiyon oluşturun. fonksiyonunuz, string içindeki her farklı karakteri anahtar olarak içeren bir sözlük oluşturmalıdır. Anahtarla ilişkilendirilen değer, bu karakterin bulunduğu stringde tüm (tamsayı) konumlarını (dizinler) içeren bir tuple olacaktır. fonksiyonunuzu oluşturduktan sonra bu sözlüğü döndürmelidir. Örnek çıktı:

```
>>> sonu = charMap ("Merhaba Dünya")
```

```
>>> yazdır (sonu)
```

```
{': (5,), 'e': (1,), 'd': (10,), 'H': (0,), 'I': (2, 3, 9), 'o ': (4, 7), 'r ': (8,), 'W ': (6,)}
```

S15- Bir stringi girdi olarak alan ve o stringdeki en yaygın 3 harfi döndüren getTopelements () adlı bir fonksiyon yazın. Karakterler aynı frekansa sahipse, alfabe sırasına göre dönmelidir.

Örneğin:

```
>>> getTopelement ( “programlama”) m
```

```
a g
```

S16- Aynı girdiyi üç kereden fazla girene kadar kullanıcıdan girdi satırlarını okuyan bir Python programı yazın.

S17- Önce bir öğrenci adı, sonra da notunu girmenizi isteyen bir program yazın. Programınız sınıfa öğrenci atamalıdır ve aynı nota sahip tüm öğrenciler aynı yere atanacaktır.

• Örneğin:

Bir öğrenci girin: Ayşe

Not girin: 50

Bir öğrenci girin: Kamuran

Bir not girin: 70

Bir öğrenci girin: Raziye

Not girin: 50

• Ve çıktı {50: [Ayşe, Raziye], 70: [Kamuran]} olmalıdır.

S18- Aşağıdaki kodun çıktısı ne olacak?

```
arr={ } arr[1]=1
```

```
arr['1']=2 arr[1]
```

```
+= 1 sum = 0
```

```
for k in arr:
```

```
sum += arr[k]
```

```
print(sum)
```

S19- İç içe de olabilen herhangi bir tuple alan ve iç içe olmayan yeni bir tuple döndüren make_simple isimli bir fonksiyon yazın.

Örneğin:

```
>>> x = (1, (2, (3,), 4), 5)
```

```
>>> make_simple(x)
```

```
(1, 2, 3, 4, 5)
```

```
>>> y = (1, 2, 3, 4)
```

```
>>> make_simple(y)
```

```
(1, 2, 3, 4)
```

S20- Bir tuple alan ve elemanlarını ters çeviren reversed_tuple () isimli bir fonksiyon oluşturun.

Eğer tuple'ın içinde de tuple olan elemanlar varsa, bu elemanlar da ters çevrilmeli.

Örneğin:

```
>>> tup = (1, (2, (3,), 4), 5)
```

```
>>> reversed_tuple(bağlantı)
```

```
(5, (4, (3,) 2), 1)
```

```
>>> y = (1, 2, 3, 4)
```

```
>>> reversed_tuple(y)
```

```
(4, 3, 2, 1)
```