ALIŞTIRMA CEVAPLARI

3. Veri Yapıları

1.

```
listem = ['Ankara', 'İzmir',
'İstanbul', 'Adana', 'Bursa',
'Kars', 'Iğdır']
listem.extend(['Edirne', 'Aydın'])
sira = listem.index('İstanbul')
listem.pop(sira)
listem.sort()
```

4. Koşullu İfadeler ve Döngüler

1.

```
for dil in prg_dilleri:
      print(dil)
yineleyici = iter(prg_dilleri)
next(yineleyici)
```

2.

```
[[i for i in range(5)] for i in
range(5)]
```

3.

```
for i in ulkeler:
    if len(i) > 5:
        print(i)
for i in range(len(ulkeler)):
    if len(ulkeler[i]) <= 5:
        ulkeler[i] = '*'
```

4.

```
dersler = ['Matematik', 'Fizik',
'Kimya']
ogretmenler = ['Cahit Arf', 'Mete
Atatüre', 'Aziz Sancar']
ders_ogretmen = zip(dersler,
ogretmenler)

for sira, cift in enumerate(ders_
ogretmen):
    print('Ders No {}: Ders:
{}, Öğretmen: {}'.format(sira,
cift[0], cift[1]))
```

5. max(x,0), x * (x > 0) ve (abs(x) + x) / 2 olası çözümlerden birkaçı.

6.

```
saat = 45
ucret = 100

if saat <= 40:
    toplam = saat * ucret
else:
    toplam = 40 * ucret + (saat -
40)*(1.5*ucret)
print(toplam)
```

7.

```
metin = 'iNcElEnecek MEtin'
sayi = 0
for harf in metin:
    if harf.isupper():
        sayi +=1
print(sayi)
sum(1 for i in metin for harf in i
if harf.isupper())
```

5. Fonksiyonlar

1.

```
def tek_cift(a):
    if a%2 == 0:
            print('{} sayısı
çifttir.'.format(a))
    else:
            print('{} sayısı
tektir.'.format(a))
```

2.

```
def ters_cevir(liste):
    ters_liste = []
    for j in range(len(liste)-1,
-1, -1):
```

```
        ters_liste.
    append(liste[j])
        return ters_liste
```

Yukarıdaki fonksiyon dışında, bir listeyi ters çevirmek için kısaca reversed() fonksiyonunu kullanabilirsiniz ya da liste[: : -1] yazabilirsiniz.

3.

```
def birlestir(a, b):
    def buyut(x):
        x = x.upper()
        return x
    a = buyut(a)
    b = buyut(b)
    return(a + b)
```

4.

```
liste_cift = (i * 2 for i in range(0, 51))
next(liste_cift)
```

5.

```
def uc_sayi_carp():
    a = input('İlk sayıyı girin: ')
    b = input('İkinci sayıyı girin: ')
    c = input('Üçüncü sayıyı girin: ')
    d = float(a)*float(b)*float(c)
    print('Girdiğiniz üç sayının çarpımı {}'.format(str(d)))
```

6.

```
def fibonacci(n):
    if n <= 0:
        print('Lütfen 0\'dan büyük bir tamsayı girin!')
    elif ((n == 1) | (n==2)) :
        x = 1
        return x
    else:
        fib = [1, 1]
        for k in range(2, n-1):
            fib.append(fib[k-1]+fib[k-2])
        x = fib[-1] + fib[-2]
        return x
```

7.

```
A: {'e': 4, 'i': 1, 'k': 3, 'l': 2, 'r': 5}
B: range(0, 10)
C: [1, 2, 3, 4, 5]
D: [1, 2, 3, 4, 5]
E: {0: 0, 1:1, 2:4, 3:9, 4:16, 5:25, 6:36, 7:49, 8:64, 9:81}
F: [[0, 0],
 [1, 1],
 [2, 4],
 [3, 9],
 [4, 16],
 [5, 25],
 [6, 36],
 [7, 49],
 [8, 64],
 [9, 81]]
```

8. Çok farklı çözümler olmakla birlikte en çok bilinen çözümlerden birisi aşağıdaki gibidir.

```
def sirala(x):
    for i in range(len(x)):
        for j in range(i+1, len(x)):
            if x[j] < x[i]:
                gecici = x[j]
                x[j] = x[i]
                x[i] = gecici
    return x
```

6. Modüller ve Nesne Yönelimli Programlama

1.

```
class Calisan:
    calisan_sayisi = 0

    def __init__(self, isim, giris_tarihi, gorev, maas):
        self.isim = isim
        self.giris_tarihi = giris_tarihi
        self.gorev = gorev
        self.maas = maas
        Calisan.calisan_sayisi += 1
    def goster(self):
        print("Adı: ", str(self.isim), "\n")
        print("İşe giriş tarihi: ", str(self.giris_tarihi), "\n")
        print("Görevi: ", str(self.gorev), "\n")
        print("Maaşı: ", str(self.maas), "\n")
    def zam_yap(self, tutar):
        self.maas += tutar
```

8. NumPy Paketi

```
np.full((4,4), fill_value=False, dtype=bool)
x = np.arange(1, 1001)
x[x % 18 == 0]
```

Örnek olarak aşağıdaki diziyi ele alalım:

```
x = np.array([4, 2, 1, 6, 9, 3, 15, 11, 10, 7])
np.sort(x)[-3:]
array([10, 11, 15])
```

4.

```
import numpy as np
def sirala(x):
    for i in range(len(x)):
        enaz = i + np.argmin(x[i:])
        (x[i], x[enaz]) = (x[enaz], x[i])
    return x
```

16. Yapay Sinir Ağları

1. Ağırlıkları öyle seçmeliyiz ki x1 ve x2 0 ise sonuç 0, diğer hallerde 1 olmalıdır. Örneğin, -50, 30, 30 ağırlıkları bu sonucu sağlayacaktır.

2.

```
import numpy as np
girdi = np.array([5,8])
agirlik = {'g11': np.array([3,-2]),
           'g12': np.array([4,0.5]),
           'g13': np.array([-1, 1]),
           'g2' : np.array([-1, 1, 2])}
katman_2 = np.array([(girdi * agirlik['g11']).sum(),
                     (girdi * agirlik['g12']).sum(),
                     (girdi * agirlik['g13']).sum()])
cikti = (katman_2 * agirlik['g2']).sum()
```

3.

```
import numpy as np
girdi = np.array([5,8])
agirlik = {'g11': np.array([3,-2]),
           'g12': np.array([4,0.5]),
           'g13': np.array([-1, 1]),
           'g2' : np.array([-1, 1, 2])}
katman_2_girdi = np.array([(girdi * agirlik['g11']).sum(),
                           (girdi * agirlik['g12']).sum(),
                           (girdi * agirlik['g13']).sum()])
katman_2_cikti = np.tanh(katman_2_girdi)
katman_3_girdi = (katman_2_cikti * agirlik['g2']).sum()
cikti = np.tanh(katman_3_girdi)
```