1. Bir dersi alan öğrencileri adi, soyadi ve o dersten aldığı vize ve final notlarını bir sözlük veri yapısında tutmak istiyoruz. Bu sözlük veri yapısında anahtarlar string türünde öğrenci numaralarından oluşurken herbir anahtara karşılık gelen değerler yine bir sözlük yapısıyla öğrenciye ait bilgiler adi, soyadi ve notlar anahtarlarında tutulmaktadır. Bu veri yapısının daha iyi anlaşılması için aşağıda bir öğrencili python_dersi isimli örnek bir sözlük sunulmaktadır. Bu bilgilere göre aşağıdaki soruları cevaplayınız.

```
python_dersi = {
   "1110310344": {
      "adi": "Ahmet",
      "soyadi": "Yagmur",
      "notlar": {
         "vize": 54,
         "final": 70
      }
   }
}
```

a. **ekle** isimli bir fonksiyon yazınız. Bu fonksiyon parametre olarak **ders** isimli sözlük türünde bir değişken, **ogr** isimli sözlük tipinde bir değişken ve son olarak **ogr_no** isimli bir değişken alacak ve yukarıda paylaşılan veri yapısına uygun şekilde bu öğrenciyi ders sözlüğüne ekleyecek.

```
def ekle(ders, ogr, ogr_no):
   ders[ogr_no] = ogr
```

b. **ortHesapla** isimli bir fonksiyon yazınız. Bu fonksiyon parametre olarak **ders** isimli sözlük türünde bir değişken ve **ogr_no** isimli bir değişken alacak ve yukarıda paylaşılan veri yapısına uygun şekilde öğrencinin aldığı vize notunun %40'ını final notununda %60'ını alıp toplayarak bir ortalama hesaplayıp bu ortalamayı döndürecek.

```
def ortHesapla(ders, ogr_no):
   ort = 0
   if ogr_no in ders:
      ogr = ders[ogr_no]
      notlar = ogr['notlar']
      vize = notlar['vize']
      final = notlar['final']
      ort = vize * 0.4 + final * 0.6
   return ort
```

c. **harfNotu** isimli bir fonksiyon yazınız. Bu fonksiyon parametre olarak **ort** isimli bir değişken alacak ve ort 90 üzerindeyse "AA", 80 ile 90 arasındaysa "BA", 70 ile 80 arasındaysa "BB", 60 ile 70 arasındaysa "CB", 50 ile 60 arasındaysa "CC" aksi taktirde "FF" değerini döndürecek.

```
def harfNotu(ort):
    hn = ""
    if ort > 90:
        hn = "AA"
    elif ort > 80:
        hn = "BA"
    elif ort > 70:
        hn = "BB"
    elif ort > 60:
        hn = "CB"
    elif ort > 50:
        hn = "CC"
    else:
        hn = "FF"
    return hn
```

d. **yazdir** isimli bir fonksiyon yazınız. Bu fonksiyon parametre olarak **ders** isimli sözlük türünde bir değişken alacak ve parametre olarak verilen ders sözlüğündeki tüm öğrencilere ait bilgileri aşağıda görüldüğü gibi yazdıracak.

Adi	Soyadi	Vize	Final	Ortalama	HarfNotu
Ahmet	Yağmur	54	70	63.6	CB
İsmail	Demir	85	70	76.0	BB
Ömer	Tekin	55	45	49.0	FF

```
def yazdir(ders):
    print("Adi\t\tSoyadi\t\tVize\t\tFinal\t\tOrtalama\t\tHarfNotu")
    for ogr_no in ders:
        ogr = ders[ogr_no]
        ad = ogr['adi']
        soyad = ogr['soyadi']
        notlar = ogr['notlar']
        vize = notlar['vize']
        final = notlar['final']
        ort = ortHesapla(ders, ogr_no)
        hn = harfNotu(ort)
        print(f"{ad}\t\t{soyad}\t\t{vize}\t\t{final}\t\t{ort}\t\t{hn}")
```

2. Aşağıdaki soruları cevaplayınız.

a. NumPy dizisi oluşturma:

- 1'den 10'a kadar olan tam sayıları içeren bir NumPy dizisi oluşturun.

```
import numpy as np
a = np.arange(1,11)
```

- `np.array([1, 2, 3, 4, 5])` şeklinde manuel bir NumPy dizisi oluşturun.

```
b = np.array([1,2,3,4,5])
```

b. Dizi özellikleri:

- `np.arange(15)` fonksiyonunu kullanarak bir NumPy dizisi oluşturun ve dizinin boyutunu, şeklini ve veri tipini ekrana yazdırın.

```
a = np.arange(15)
print(f"Dizinin boyutu = {a.ndim}")
print(f"Dizinin şekli = {a.shape}")
print(f"Dizinin veri tipi = {type(a)}")
```

c. Temel işlemler:

- 1'den 5'e kadar olan sayılardan oluşan iki NumPy dizisinin eleman bazında toplamını yazdırın.

```
a = np.arange(1,6)
b = np.arange(1,6)
print(a + b)
```

- Aynı dizilerle eleman bazında çarpım işlemi yapın ve yazdırın.

```
print(a * b)
```

d. Dilimleme ve indeksleme:

- 10 elemanlı rastgele bir NumPy dizisi oluşturun ve bu dizinin ilk 5 elemanını seçin.

```
x = np.random.rand(10)

print(x[:5])
```

- 5x5 boyutlarında rastgele sayılardan oluşan bir NumPy matrisi oluşturun ve bu matrisin köşegenindeki (diagonal) elemanları bulun.

  A = np.random.rand(5,5)

  print(np.diag(A))

e. Şekil değiştirme (reshaping):

- 1'den 12'ye kadar olan sayıları içeren bir NumPy dizisini 3x4 boyutlarında bir matrise dönüştürün.

  x = np.arange(1,13)

  x = x.reshape(3,4)

- Bu matrisi 4x3 boyutunda yeni bir matrise yeniden şekillendirin.

  x = x.reshape(4,3)

f. Matematiksel işlemler:

- 5x5 boyutlarında rastgele sayılardan oluşan bir NumPy matrisi oluşturun ve her sütunun ortalamasını hesaplayın. Satır ortalaması ?

  A = np.random.rand(5,5)

  A.mean(axis=0)

- Aynı matrisin her bir elemanının karesini alın.

  A ** 2

g. Gelişmiş indeksleme ve maskeleme:

- 10 elemanlı rastgele bir NumPy dizisi oluşturun ve bu dizinin 5'ten büyük olan elemanlarını bulun.

  x = np.random.randint(0,10,10)

  x[x>5]

- Aynı dizi üzerinde, çift olan elemanları seçin ve bunları ekrana yazdırın.

  print(x[x % 2 == 0])

h. Lineer cebir işlemleri:

- 2x2 boyutlarında iki NumPy matrisi oluşturun ve bu matrislerin çarpımını gerçekleştirin.

  A = np.random.rand(2,2)

  B = np.random.rand(2,2)

  A.dot(B)

- Aynı matrisler için tersini (inverse) hesaplayın. Determinant ?

  print(np.linalg.inv(A))

  print(np.linalg.inv(B))

i. Rastgele sayı üretimi ve istatistiksel hesaplamalar:

- 1000 elemanlı, normal dağılıma sahip bir NumPy dizisi oluşturun ve bu dizinin ortalamasını ve standart sapmasını hesaplayın.

  x = np.random.randn(1000)

  print(x.mean())

  print(x.std( ))

- Bu dizinin histogramını çizdirin.

  import matplotlib.pyplot as plt

  plt.hist(x);

j. Aşağıdaki lineer denklem sistemini çözmek için NumPy kullanın:

x + y + z = 6
2x + 5y + z = -4
2x + 3y + 8z = 10

```
A = np.array([
   [1, 1, 1],
   [2, 5, 1],
   [2, 3, 8]
])

B = np.array([6, -4, 10])

solution = np.linalg.solve(A, B)
print(solution)
```