**S1.** Aşağıda sözde kodu verilen program için Python kodunu yazın?

|  |  |
| --- | --- |
| *kullanıcıdan bir tamsayı girdisi iste*  *girişi oku ve onu bir tam sayıya dönüştür*  *tamsayı 0 değilken*  *tam sayı 100 veya daha fazlaysa "sıcak" yazdırın*  *tam sayı 60 veya daha az ise, "soğuk" yazdırın*  *aksi takdirde "değeri" yazdırın*  *kullanıcıdan bir tamsayı girdisi isteme*  *girişi oku ve onu bir tam sayıya dönüştür* | while 1:  girdi=*int*(input("Bir tamsayi giriniz:"))  if girdi != 0:  if girdi >= 100:  print("Sicak")  elif girdi <= 60:  print("Soguk")  else:  print(girdi) |

**S2.** Python'da yerel değişkenler ve global değişkenleri bir örnek ile açıklayınız?

|  |
| --- |
| a=10  *def* yazdir():  b=20  #a global bir değişkendir ve fonksiyon içerisinden erişilebilir  print("Yazdir fonksiyonu ici (a): ",a)  #b yerel bir değişkendir ve sadece tanımlandığı alan yani fonksiyon içerisinden erişilebilir  print("Yazdir fonksiyonu ici (b): ",b)  yazdir()  #a değişkeni global olduğu için erişilebilir  print("Global alan (a):",a)  #buradaki tanımlama hatalıdır çünkü b değişkeni yazdir fonsiyonu içerisinde tanımlanmıştır ve dışında kullanılamaz  #print(b) |

**S3.** Python yorumlayıcı dil olmasından dolayı dinamik tip denetimi vardır. Derleyici diller ise statik tip denetimi vardır. Bu iki yaklaşımı Python’ın avantajını örnek program yazarak açıklayın?

|  |
| --- |
| #String(katar) türünde tanımlanan bir değer değişkene atanır ve Python yorumlayıcısı bunun türünü otomatik olarak algılar  a="Ahmet"  #Integer(tam sayi) türünde tanımlanan bir değer değişkene atanır ve Python yorumlayıcısı bunun türünü otomatik olarak algılar  b=5  #Float(tgerçel sayi) türünde tanımlanan bir değer değişkene atanır ve Python yorumlayıcısı bunun türünü otomatik olarak algılar  c=3.2  #Boolean(doğru/yanlış) türünde tanımlanan bir değer değişkene atanır ve Python yorumlayıcısı bunun türünü otomatik olarak algılar  d=True  #tip değerlerini(sınıf) ekranda görebilmek için type fonksiyonu kullanılarak tip değeri alınmak istenen değişken parametre olarak verilir  print("Deger:",a,"\tTip:",*type*(a)) #a değişkeni str sınıfına dahildir yani bir string ifadedir  print("Deger:",b,"\tTip:",*type*(b)) #b değişkeni int sınıfına dahildir yani bir integer ifadedir  print("Deger:",c,"\tTip:",*type*(c)) #c değişkeni float sınıfına dahildir yani bir float ifadedir  print("Deger:",d,"\tTip:",*type*(d)) #d değişkeni bool sınıfına dahildir yani bir bool ifadedir  #Derlemeli dillerde (örnek C dili) değişkenin türü önceden belirtilmelidir, python programlama dilinde dinamik olduğundan ve  #otomatik olarak belirlendiğinden esneklik sağlamaktadır |

**S4.** Varsayım olarak iPhone şifresi, her biri 0 ile 9 arasında 4 basamak uzunluğundadır. Başka birinin, olası bir şifreyi (tam olarak 4 karakter uzunluğunda) temsil eden tek bir dize bağımsız değişkenini alan bir *test\_password* işlevi yazdığını varsayalım. *test\_password*, giriş şifresi doğru ise telefonun kilidini açar, aksi takdirde false değeri döndürür. Hiçbir argüman almayan ve telefonun kilidini açan parolayı temsil eden bir dize döndüren bir Python işlevi crack\_iphone yazın?

|  |
| --- |
| *def* crack\_iphone():  for sayi in range(0,10000):  hane\_sayisi=len(*str*(sayi))  eklenecek\_adet=4-hane\_sayisi  eklenecek\_sifir=eklenecek\_adet\*"0"  yeni\_sayi=eklenecek\_sifir+*str*(sayi)  if test\_password(yeni\_sayi) != False:  print("Sifre:",yeni\_sayi)  return |

**S5.** A, m ve n olmak üzere üç bağımsız değişken alan sum\_elements adlı bir Python işlevi yazılacaktır. A, ondalıklı değerler alan 2 boyutlu bir numpy dizisidir. m, matristeki satır sayısıdır ve n sütun sayısıdır. İşleviniz, girdi dizisinin tüm öğelerinin toplamını döndürmelidir. İşlev saf Python yetenekleri ile yazılacağı için herhangi bir modül kullanılmamalıdır?

|  |
| --- |
| *def* sum\_elements(*A*,*m*,*n*):  toplam=0  for satir in range(m):  for sutun in range(n):  toplam = toplam + A[satir][sutun]  return toplam |

**S6.** Import kullanmadan, iki sözlük argümanı dictA ve dictB alan *print\_pass* adlı bir fonksiyon yazılacaktır. dictA, bir öğrenci Id'si ile adı arasında bir eşleştirme içeren bir sözlüktür. Örneğin,

dictA = {"AVeli": "Ali Veli", "OBulbul": "Osman Bülbül", "ZTekin": "Zeynep Tekin"}

dictB, bir öğrencinin Id'si ile onun notu arasındaki bir eşlemeyi içeren bir sözlüktür.

dictB = {" AVeli ": 85, " OBulbul ": 90, " ZTekin”: 55}

70 veya üzeri puan alan öğrencilerin adlarını ve notlarını yazdıran *print\_pass* adında Python işlevi yazın. Örnek çıktı:

Ali Veli 85

Osman Bülbül 90

|  |
| --- |
| *def* print\_pass(*dictA*,*dictB*):  for anahtar in dictB:  if dictB.get(anahtar) >= 70:  print(dictA.get(anahtar),dictB.get(anahtar))  dictA = {"AVeli": "Ali Veli", "OBulbul": "Osman Bülbül", "ZTekin": "Zeynep Tekin"}  dictB = {"AVeli": 85, "OBulbul": 90, "ZTekin": 55}  print\_pass(dictA,dictB) |