

VBM651 2. Vize

27 Aralık 2022 – 1 Ocak 2023

Soru 1:

Elimizde iki adet zıplayan robot bulunmaktadır. İkisi de bir başlangıç noktasından aynı istikamete doğru zıplayarak ilerleyecektir.

- Birinci robot **x1** konumundan başlar ve **v1** metre kadar ileri zıplar.
- İkinci robot **x2** konumundan başlar ve **v2** metre kadar ileri zıplar.

Amacımız, aynı zamanda zıplamaya başlayan robotların bir t anında aynı noktaya gelip gelmeyeceklerini tespit etmektir (True/False).

Örnek:

$$x1 = 2 \quad v1 = 1$$

$$x2 = 1 \quad v2 = 2$$

Bir zıplama sonrası, ikisinin de konumu $x = 3$ olacaktır. Cevap **True**.

Örnek2:

$$x1 = 0 \quad v1 = 3$$

$$x2 = 4 \quad v2 = 2$$

Cevap **True**.

Örnek 3:

$$x1 = 0 \quad v1 = 2$$

$$x2 = 5 \quad v2 = 3$$

Cevap **False**.

Yukarıdaki problem çözen “robojump” fonksiyonunu yazınız.

def robojump(x1, v1, x2, v2):

Soru 2:

Bir yarışma programı için fonksiyon yazacağız. Fonksiyonun amacı okuduğu “yarisma.txt” içinde bulunan puanlara göre yarışmanın kazananını belirlemek.

Örnek girdi kütüğü: Üç satır(3 yarışmacı) ve her bir satırda dört adet değer (4 ayrı puan) vardır.

40	10	5	5
30	20	10	0
30	30	10	5

1. Yarışmacı 1. Soruda en yüksek puanı almıştır. (40-30-30)
3. Yarışmacı 2. Soruda en yüksek puanı almıştır. (10-20-30)
2. ve 3. Yarışmacı 3. Soruda en yüksek puanı almışlardır (5-10-10)
1. ve 3. Yarışmacı 4. Soruda en yüksek puanı almışlardır (5-0-5)

Yarışma sonunda

- 1. Yarışmacı: 2 puan
- 2. Yarışmacı: 1 puan
- 3. Yarışmacı: 3 puan

Almıştır. Kazanan 3. Yarışmacıdır.

Kısıtlar:

Satır sayısı değişken olabilir(1'den 100'e kadar).

Satır içindeki değerler değişken olabilir (1'den 100'e kadar), fakat her satırda aynı olacaktır.

Yarışma sonunda birden fazla kazanan olabilir. Aynı en yüksek puanı alanlar kazanmış sayılır.

Veri kütükten okunup, sonuç ekrana yazdırılacaktır.

Contest fonksiyonu kütük adını parameter olarak alacaktır.

def contest(filename):

Soru 3: Bir video oyununda oyun sonu başarı listesi tutulmaktadır. Başarı listesi aşağıdaki düzende belirlenmektedir.

- En yüksek puanı alan kişi 1. Sırada yer alır.
- Eğer aynı puanı almış oyuncular varsa aynı sıralamayı alırlar. Sonraki oyuncular da ona göre sıralamada bulunurlar.

Örnek:

Oyuncu ve Puanı	Sıralama
Oyuncu A 100	1
Oyuncu B 90	2
Oyuncu C 90	2
Oyuncu D 50	3

Sizden istenen mevcut başarı listesi ve yeni oyuncunun birkaç oyun sonra aldığı puanların olduğu listeyi parametre olarak alan bir fonksiyon yazmanızdır. Fonksiyon oyuncunun her puanı için elde ettiği sıraları hesaplayıp, bunları ekrana yazdıracaktır.

Örnek:

ranklist = [100, 100, 50, 40, 40, 20, 10] # mevcut başarı listesi, azalan sırada verilecektir.
scores = [5, 25, 50, 120] # oyuncunun aldığı puanlar, artan sırada verilecektir.

Ranklist ilk hali:

Oyuncu ve Puanı	Sıralama
100	1
100	1
50	2
40	3
40	3
20	4
10	5

Oyuncunun ilk puanı = 5, -> sıralaması 6

Oyuncu ve Puanı	Sıralama
100	1
100	1
50	2
40	3
40	3
20	4
10	5
Yeni oyuncu- 5	6

Oyuncunun ikinci puanı = 25, -> sıralaması 4

Oyuncu ve Puanı	Sıralama
100	1
100	1
50	2
40	3
40	3
Yeni oyuncu- 25	4
20	5
10	6

Oyuncunun üçüncü puanı = 50, -> sıralaması 2

Oyuncu ve Puanı	Sıralama
100	1
100	1
50	2
Yeni oyuncu- 50	2
40	3
40	3
20	4
10	5

Oyuncunun dördüncü puanı = 120, -> sıralaması 1

Oyuncu ve Puanı	Sıralama
Yeni oyuncu- 120	1
100	2
100	2
50	3
40	4
40	4
20	5
10	6

Ekran çıktısı:

6

4

2

1

Kısıtlar:

- Ranklist içinde en az bir değer olacaktır. En fazla 100 değer olacaktır.
- Scores içinde en az bir değer olacaktır. En fazla 50 değer olacaktır.

Yukarıda bahsedilen işlemi gerçekleştiren fonksiyonu yazınız.

def scoreboard(ranklist, scores):