

Hafta 3 – Hard Task

Hakkında:

Bu ödevde sizden beklenen, sürücülerini en uygun konumlara yönlendiren bir otopark otomasyonu yapmanızdır. Bu otomasyon aşağıdaki kurallara ve şartlara uygun olarak yapılmalıdır.

Ödevin Kazanımları:

- 2 Boyutlu listelerin kullanımı, döngüde çalışması ve elemanların indexlenmesi,
- Fonksiyonların oluşturulması ve kullanımı,
- Recursion ile işlem yapması,
- Bir projenin aşama aşama düzenli bir şekilde oluşturulması ve birleştirilmesi

Detaylar:

- 1) Amacımız, öncelikle $n \times n$ boyutlarında, tüm elemanları 0 olan bir matris (2D liste) oluşturmaktır. Bu listedeki her bir eleman, 0 veya 1 değerlerinden birini alacaktır. 0 değeri yerin boş olduğunu, 1 ise dolu olduğunu göstermektedir. Sonrasında, yazılacak fonksiyonlarla her giren araç, o an otoparktaki en uygun yere göre yerleştirilecektir.
 - Bu projede uygunluk kapıya uzaklıkla doğru orantılıdır. Uzaklığı ise, elemanın satır ve sütun değerlerinin toplamı olarak tanımlayınız.
 - Yani yeni araç, o an boş olan $i + j$ toplamı en yüksek yere yönlendirilmelidir.
 - Bu toplam değeri aynı olan yerler için yerleştirme sırası önemsizdir.
- 2) Kodun ana şeması sizin için yazılmıştır. Yapmanız gereken aşağıdaki kodu kopyalayıp dosyanıza yapıştırmak, gerekli fonksiyonları hazırlamak ve kodun istediği gibi çalıştığından emin olmaktır. **main() fonksiyonunu ve diğer kısımlarda değişiklik yapmamalısınız, sadece aradaki fonksiyonları düzenlemeniz gerekiyor.**

```
def main():
    n = int(input("Otoparkın Bir Kenar Uzunluğu: "))
    otopark = otopark_olustur(n)
    print(f"Otopark:\n{otopark_ciz(otopark)}")
    while True:
        try:
            cevap = int(input("Eklenecek Araç Sayısı Girin: "))
            arac_ekle(otopark, cevap)
            print(f"Otopark:\n{otopark_ciz(otopark)}")
        except ValueError:
            print("Geçerli bir değer girilmedi! İşlem iptal ediliyor..")
            break
        except:
            print("Hata! İşlem iptal ediliyor...")
            break
```

```
def otopark_olustur(n):  
    ...  
def otopark_ciz(otopark):  
    ...  
def yer_bul(otopark):  
    ...  
def arac_ekle(otopark, n):  
    ...  
  
main()
```

3) Fonksiyonların anlamları ve kullanımları aşağıdaki tanımlara uygun olmalıdır:

- otopark_olustur(n): Uygulamanın başında bir kez çalışarak otoparkın oluşumunu sağlar. nxn'lik bir otopark 2D listesi oluşturur. Bu otopark değerini döndürür.
- otopark_ciz(otopark): Parametre olarak verilen otoparkın görselleştirilmesini sağlar. Otopark listesini alır, satırların alt alta sıralandığı bir string döndürür.
+ İpucu: String değerlerde alt satıra geçmek için “\n” operatörünü kullanınız.
- yer_bul(otopark): O anki otopark değerini parametre olarak alır, o anki en uygun konumun kordinatlarını bir (x, y) tüple ikilisi olarak verir.
+ En uygun konum, o an boş olan yerler arasında i+j toplamı en yüksek olandır.
- arac_ekle(otopark, n): O anki otopark değeri ve kullanıcıdan alınan n değerini parametre alır, otoparka sırayla n adet araç girişi yapar.
+ Araçların yerleştirilmesinde konumlara karar vermek için yer_bul() fonksiyonunu kullanmanız beklenmektedir.
+ Bu fonksiyonun içinde, for ve while döngülerinin kullanımı yasaktır.
+ Birden fazla araç girişi yapılacağı durumlarda, recursion yöntemi kullanmanız beklenmektedir.

4) Fonksiyonları doğru bir şekilde tanımladığınız durumda, program eksiksiz olarak çalışacaktır.

Fonksiyonları düzenlemek dışında, main fonksiyonunu değiştirmemeniz, başka fonksiyonlar yazmamanız vb. bir değişiklik yapmamanız beklenmektedir.

5) Programın sonunda, örneklerde olduğu gibi, “main()” komudu ile main fonksiyonunu çağırmayı unutmayınız.

6) Projeyi tamamladıktan sonra kodunuz buna benzemelidir:

```
main.py X
main.py > main
1 def main():
2     n = int(input("Otoparkın Bir Kenar Uzunluğu: ")) # Kullanıcıdan Otopark Büyüklüğünün Alınması
3     otopark = otopark_olustur(n) # Otoparkın oluşturulması
4     print(f"Otopark:\n{otopark_ciz(otopark)}") # Otoparkın Görselleştirilmesi
5     while True: # Ekleme İşleminin Defalarca Yapılması
6         try:
7             cevap = int(input("Eklenecek Araç Sayısı Girin: ")) # Eklenecek "n" Araç Sayısının Alınması
8             arac_ekle(otopark, cevap) # Otopark'a "n" Araç Eklenmesi
9             print(f"Otopark:\n{otopark_ciz(otopark)}") # Otoparkın Görselleştirilmesi
10            except ValueError: # Kullanıcı Int Değer Girmezse Oluşacak Hata
11                print("Geçerli bir değer girilmedi! İşlem iptal ediliyor..")
12                break
13            except: # Diğer Hataların Ayıklanması
14                print("Hata! İşlem iptal ediliyor...")
15                break
16
17 def otopark_olustur(n):
18     """
19     nxn'lik bir otopark 2D listesi oluşturur. Liste halinde döndürür.
20     """
21     ...
22
23 def otopark_ciz(otopark):
24     """
25     Otoparkı, her satırı alt alta gelecek şekilde görselleştirir. String olarak döndürür.
26     """
27     ...
28
29 def yer_bul(otopark):
30     """
31     Otoparkta en uygun yeri döndürür. En uygun konum, o an müsait olan kapıya en uzak noktadır. (i ve j kordinatlarının toplamı)
32     Konumu, (i, j) tuple ikilisi olarak döndürür
33     """
34     ...
35
36 def arac_ekle(otopark, n):
37     """
38     Otoparka "n" adet araç ekler. Bunu "yer_bul()" fonksiyonunu kullanarak yapar. Bu kısımda loop değil recursion kullanmanız
39     beklenmektedir. Otoparkın araçlar eklenmiş halini bir 2D dizi olarak döndürür.
40     """
41     ...
42
43 main()
```

7) Projeyi tamamladıktan sonra örnek konsol çıktınız bu şekilde olmalıdır:

```
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
C:\Users\UmurAta\Masaüstü\hard_task
Otoparkın Bir Kenar Uzunluğu: 4
Otopark:
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]

Eklenecek Araç Sayısı Girin: 3
Otopark:
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 1]
[0, 0, 1, 1]

Eklenecek Araç Sayısı Girin: 1
Otopark:
[0, 0, 0, 0]
[0, 0, 0, 1]
[0, 0, 0, 1]
[0, 0, 1, 1]

Eklenecek Araç Sayısı Girin: 2
Otopark:
[0, 0, 0, 0]
[0, 0, 0, 1]
[0, 0, 1, 1]
[0, 1, 1, 1]

Eklenecek Araç Sayısı Girin: 5
Otopark:
[0, 0, 1, 1]
[0, 0, 1, 1]
[0, 1, 1, 1]
[1, 1, 1, 1]

Eklenecek Araç Sayısı Girin: exit
Geçerli bir değer girilmedi! İşlem iptal ediliyor..
```