

## 1. Soru 1

İlk soru için aşağıda bulunan kodu kopyalayıp kullanarak program için gerekli olan iki boyutlu diziyi oluşturunuz.

İki boyutlu dizi de bulunan bilgiler hata kodları olarak düşünülebilir. İlk hatanın majör değeri ikincisi minör değeri olarak düşünülmelidir. İlk soru için sizden istenen sadece kullanıcıdan bu bilgiyi ne şekilde sıralanmasının istendiğidir. Ve bu sıralama isteğine göre değerler sıralanıp ekrana basılacaktır.

Sıralama seçeneği 1 → En kritik hatadan en az kritik hataya göre sıralanma

Sıralama seçeneği 2 → En az kritik hatadan en kritik hataya

Sıralama seçeneği 3 → En sık rastlanan hatadan en nadir rastlanan hataya

Sıralama seçeneği 4 → En nadir rastlanan hatadan en çok karşılaşılan hataya

```
private static int[,] ArrayInitializer()
{
    return new int[,] {
        { 5, 6 }, { 1, 2 }, { 1, 3 }, { 2, 2 }, {11, 7 }, { 5, 3 }, { 4, 11},
        {15, 8 }, {14, 2 }, { 3, 9 }, { 7, 4 }, { 6, 8 }, { 8, 6 }, { 9, 5 },
        {11, 3 }, {15, 5 }, {13, 15}, {18, 14}, { 5, 19}, {15, 16}, {15, 11},
        {13, 12}, {14, 5 }, { 1, 13}, { 8, 5 }, { 9, 7 }
    };
}
```

## 2. Soru 2

Bu uygulama aracılığı ile kişilere yazılmış cezalar ve ödeme bilgileri üzerinden sıralamalar yapılacaktır.

İkinci soruda kullanacağınız listeyi aşağıdaki fonksiyonu kullanarak elde edebilirsiniz.

İlk boyut cezalara temsil edecektir. Her bir ceza id değerine sahip ilk bilgi ile başlayacaktır. İd değeri eşsiz bir değer taşımaktadır. Bunu kimlik numarası gibi düşünebilirsiniz. İkinci değer ödenmiş miktar üçüncü değer ise toplamda alınmış cezadır.

Bir kişi birden fazla ceza almış olabilir. Bu bilgiler göz önüne alınarak ve kullanıcıdan alınan istek doğrultusunda

Sıralama seçeneği 1 → Ceza alma sayısı en çok olan kişiden en az olana doğru kimlik numaralarını ve aldıkları ceza sayılarını ekrana yazınız.

Sıralama seçeneği 2 → Toplamda en fazla para cezası alandan en az alan kişiye doğru sıralayınız ve aldıkları toplam para cezasını ekrana yazınız.

Sıralama seçeneği 3 → Anlık olarak bakılınca ödenmeyi bekleyen borç miktarı en fazla olan kişiden en az olana doğru sıralayınız ve ne kadar borçlarının olduğunu ekrana yazınız.

Sıralama seçeneği 4 → numaralı operasyon olarak kişileri kimlik numaralarına göre sıralayınız ve yanlarına hem aldıkları cezaların toplam tutarını hem ödedikleri toplam tutarı hem kalan ödemeleri ve kaç adet cezaları olduğunu yazınız.

Adem : 500 370 130 (5 ceza)

```
private static int[,] ArrayInitializer()
{
    return new int[,] {
        { 1, 10, 10 }, { 2, 10, 20 }, { 3, 15, 100 }, { 4, 250, 300 },
        { 1, 50, 150 }, { 6, 47, 60 }, { 7, 50, 150 }, { 3, 0, 175 },
        { 3, 80, 145 }, { 8, 0, 250 }, { 9, 15, 45 }, { 4, 40, 40 },
        { 6, 8, 15 }, { 8, 60, 60 }, { 10, 50, 50 }, { 11, 451, 452 },
        { 12, 15, 46 }, { 13, 45, 55 }, { 13, 50, 95 }, { 14, 55, 80 },
        { 10, 20, 50 }, { 15, 16, 46 }, { 16, 0, 450 }, { 17, 10, 100 },
        { 17, 5, 145 }, { 2, 13, 23 }, { 8, 95, 235 }, { 9, 70, 70 }
    };
}
```

### 3. Soru 3

Bu soru için permütasyon bilgilerinizi hatırlamanız gerekmektedir.

Sorunun anlaşılması için bir örnek üzerinden soru anlatılacaktır. Sorunun çözümü sırasında kod akışı kısmını inceleyiniz.

#### Örnek :

Sayısal loto için 1-49 arasında 6 adet sayı seçilebilmektedir. Bu da demek oluyor ki  $\binom{49}{6}$  farklı seçim yapılabilmektedir. Fakat loto oynayacak kişi 2 adet şanslı sayı belirler ve bunları her loto oynadığı zaman kullanırsa oluşabilecek farklı seçim sayısı  $\binom{47}{4}$  şeklinde olacaktır. Veya eğer 3 şanssız sayı belirler ve bunlarla hiç oynamazsa seçim sayısı  $\binom{46}{6}$  şeklinde belirlenebilmektedir.

#### Kod akışı:

- Kullanıcıdan maximum ve minimum değerler alınacaktır (Maximum değer minimum değerden büyük olduğunu kontrol ediniz)
- Kaç adet sayı seçileceğini kullanıcıdan alınız ( 51 - 70 arasında sayılar seçilecekse 20den daha az sayı seçilecektir ki şans faktörü oluşsun)
- Bir önceki adım tamamlandıktan sonra kaç farklı seçim yapılabileceği ekrana yazılacaktır. (51-70 arasında seçim yapılacak ve 2 sayı seçilecekse 380 farklı seçim yapılabilir)
- Daha sonra kaç defa oynamak istediğini kullanıcıya soracağız. (Bir önceki durum göz önüne alındığı zaman 380 defadan fazla oynayamaz)
- Kaç defa oynamak istediğini de öğrendikten sonra bir fonksiyon aracılığı ile kaç tane şanslı sayı girebileceği hesaplayıp ekrana bu bilgiyi gösteriniz.
- Kullanıcıya şanlı numarası varsa girmesini yoksa enter tuşuna basarak devam etmesini söyleyiniz
  - Eğer sayı girerse bu sayı saklanacak ve ekrana kaç tane daha şanslı numara girebileceği söylenecek ve bu işlem yenilenecek
  - Eğer kullanıcı enter tuşuna basarsa bir sonraki adıma geçebilirsiniz
- Şanssız numara girme hakkı varsa, kullanıcıya şanssız numarası varsa girmesini yoksa enter tuşuna basarak devam etmesini söyleyiniz
  - Kullanıcı şanssız numarasını girerse bu sayı saklanacak ve kaç tane daha şanssız numara girebileceği ekrana basılacak
  - Kullanıcı şanssız numara girmezse bir sonraki adıma geçebilirsiniz
- Son adım olarak oluşturulan kuponlar ekrana yazılacak