

- 1) Kullanıcıdan, kullanıcının girdisi "bitti" dizgisi olana kadar sırayla dizgiler okuyan ve okuduğu dizgilerdeki (son okuduğu "bitti" dizgisi hariç) ortalama karakter sayısını çıkıtlayan tam bir C# programı yazın. Örneğin okunan dizgiler sırasıyla, "blm236", "nesneye", "dayali", "programlama", "final", "sinavini", "da", "sonunda", "verdik", "yasasin", "bitti" olursa okunan dizgilerin toplam karakter sayısı  $6 + 7 + 6 + 11 + 5 + 8 + 2 + 7 + 6 + 7 = 65$  olur. Toplam on (10) dizgi okunduğu için, çıkıtlanması gereken değer  $65/10 = 6.50$  olacaktır.

*Not:* Herhangi bir dizgiyi standart girdi aracından (klavyeden) okumanın yolu `str = System.Console.ReadLine();` kodunu kullanmaktır:

- 2) Aşağıda eksik tanımlı verilen `kesir` sınıfı için ikili (binary) + işlecinin üstüne yüklemek (operator overloading) istiyoruz. Sınıf tanımındaki `pay` üye değişkeni kesirin payını, `payda` üye değişkeni de paydasını göstermektedir. + işlecini kullanarak iki `kesir` nesnesini toplananın (`k1 + k2`) başka bir `kesir` nesnesi

(k3) üretmesini sağlayacağız. Bu durumda kesirlerin paydasını eşitlemek ve doğru işlemleri yapmak gerekiyor. + işlecinin üstüne yükleme işini başaran fonksiyonun kodunu kutuya yazın.

Örnek: k1 kesirinin pay değeri 3, payda değeri 5; k2 kesirinin pay değeri 4, payda değeri 14 olduğunda k3 (k1 + k2) kesirinin pay değeri 62, payda değeri 70 olur. Üstüne yüklenen + işleci yapılabilecek sadeleştirmeleri dikkate almayacaktır.

```
class kesir {
    int pay, payda;
    public kesir() {pay = payda = 1;}
    public kesir (int py, int pyd) {
        pay = py;
        payda = (pyd == 0) ? 1 : pyd;
    }
}
```

```
}
```

- 3)** Değişken sayıda `int` tipinde parametre alabilen ve parametrelerinin tek olanlarından bir dizi oluşturarak bu diziyi dönen `tekler` adlı bir fonksiyon yazın (C#'ta C'den farklı olarak dizilerin de fonksiyonlardan dönülebildiğini hatırlayalım). Fonksiyon herhangi bir sınıf tipi tanımında yer alan bir üye metot olabilir. Örneğin,

orneksinif adlı sınıfın public bir metodu olabilir. Bu durumda orneksinif tipinde bir nesnenin başvurusunun adı ob ise, fonksiyon ob.tekler(...?...) diye çağırılır. Örnek olarak, fonksiyona verilen parametrelerin değerleri sırasıyla 7, 20, 60, 111, 14, 65, 90, 125 olursa (değişken sayıda parametrenin fonksiyona nasıl verilebileceği hakkındaki kararı siz verecek ve fonksiyon kodunu ona göre yazacaksınız), fonksiyondan dönülmesi gereken tamsayı dizisi [7, 111, 65, 125] olmalıdır.

- 4) Aşağıda görülen, kök düğümde 12 değeri bulunan ikili ağacı (binary tree) artsıra gezme (postorder traversal) kullanarak elemanları işleyin. İşleme, işlenen düğümün başlangıçta boş olan bir ikili arama ağacına (BST = binary search tree) yerleştirilmesi şeklinde olmalı. Her düğüm işlendikçe, oluşturulan BST büyümeli ve en sonunda 13 düğümü olan bir BST oluşmalı. Son olarak, oluşan BSTyi içsıra gezme (inorder

traversal) kullanarak işleyin. Buradaki işleme, düğümlerdeki değerleri görülme sırasına göre yazmak olsun. Bu soruda kod yazmanız beklenmiyor. İstenen, oluşan BSTyi resim çizerek göstermeniz (aşağıda verilen ikili ağaca benzer) ve son aşamada yapılan içsıra gezmede elemanları doğru sırayla yazmaktan ibaret.

