

(1.ÖĞRETİM) NESNE TABANLI PROGRAMLAMA-1 FİNAL SINAVI SORULARI

**UYARILAR:**

- \* 1,2,3,4.Soruların cevaplarında "**if, switch, for, while, do-while, foreach**" yazılmayacaktır.
- \* 1,2,3,4.Soruların cevaplarında '\*', '/', '%' işaretleri kullanılmayacaktır.
- \* Tüm metotlarda ve özelliklerin iç kısımlarında **sadece bir satır kod** yazılacaktır.
- \* 1,2,3.Sorularda sadece metotlar yazılacaktır.(Main metodu içerisi yazılmıştır)
- \* 4.Soruda sadece sınıf, metotlar ve özellikler yazılacaktır.(Main metodu içerisi yazılmıştır)
- \* 1,2,3,4.Sorularda metot ve özellik isimleri "Main" metodundan çağrıldığı gibi tanımlanacaktır.
- \* 1,2,3.Sorularda (4.soru hariç) metotlar **ÖZYİNELEMELİ (RECURSIVE)** olarak yazılacaktır.

**1.Soru:** Farklı miktarda sayının toplanmasını sağlayan metodu yazınız.

**Bilgi:** Dizi uzunluğunu elde etmek için "Length" özelliği kullanılır. (dizi.Length)

```
int adet = 0, toplam = Topla(_____ adet, 11, 22, 33, 44, 55, 66, 77, 88, 99);
```

```
Console.WriteLine("{0} adet sayının toplam:{1}", adet, toplam); //9 adet sayının toplamı:495
```

**2.Soru:** Aşağıdaki matematik sorusunun çözümünü yapan metotları oluşturunuz.

**Problem:** "Dört yanı duvarlarla çevrili bir yere **bir çift tavşan** konmuştur. Her çift tavşanın **bir ay** içinde **eni bir çift** tavşan yavruladığı, her yeni çiftin de erginleşmesi için **bir ay** gerektiği ve tavşanların ölmediği varsayılırsa, dört duvarın arasında **en az 500 çift** tavşanın olması için kaç ay geçmesi gerekir?"

**Çözüm:** Tavşan çiftlerinin sayısı aylara göre şu sıralamayı ortaya koymaktadır:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610,....

İlk iki sayı hariç, her sayı kendisinden önce gelen iki sayının toplamına eşittir.

```
int ay = Ay(500), tavşan = Tavşan(ay); //çift olarak hesaplandı
```

```
Console.WriteLine("Ay: {0}, Tavşan: {1}", ay, tavşan); //Ay:15, Tavşan: 610
```

**3.Soru:** **Üç basamaklı en küçük asal** sayıyı bulan metodu yazınız.

**Bilgi:** Asal sayılar,kendisinden ve 1 sayısından başka böleni olmayan, 1'den büyük pozitif tam sayılar biçiminde tanımlanmaktadır.Yüzden küçük asal sayılar:

2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97

```
Console.WriteLine("Üç Basamaklı En Küçük Asal Sayı : {0}", EnKüçük_AsalSayı(100, 999)); //101
```

**4.Soru:** Aşağıda isimleri ve sayısal notları girilen derslere göre sınıfı oluşturunuz.

\* Sınıf adı : Ders

\* Özellikler: Ad(metin), Not(tamsayı, sadece 1,2,3 değerleri girilebilir)

\* Metotlar : HarfNot (Not özelliğine göre 1 ise "C", 2 ise "B", 3 ise "A" sonucu döndürecek)

ToString (Ekran çıktısı "Fizik : 1 = C" şeklinde olacak)

```
Console.WriteLine(new Ders() { Ad = "Fizik", Not = 1 });
```

```
Console.WriteLine(new Ders() { Ad = "Kimya", Not = 2 });
```

```
Console.WriteLine(new Ders() { Ad = "Tarih", Not = 3 });
```

9 adet sayının toplamı : 495

Ay: 15, Tavşan: 610

Üç Basamaklı En Küçük Asal Sayı : 101

Fizik : 1 = C

Kimya : 2 = B

Tarih : 3 = A

Tüm soruların ekran çıktısı:

Süre : 40 dakika

(1.ÖĞRETİM) NESNE TABANLI PROGRAMLAMA-1 FİNAL SINAVI CEVAPLARI

---

//1.soru:

```
static int Topla(ref int adet, params int[] Sayılar) (10p)
{
    return adet == Sayılar.Length ? 0 : Sayılar[adet++] + Topla(ref adet, Sayılar); (15p)
}
```

//2.soru:

```
static int Ay(int adet) (5p)
{
    return Ay(0, adet);
}
static int Ay(int a, int b) (10p)
{
    return Tavşan(a) >= b ? a : Ay(a + 1, b);
}
static int Tavşan(int ay) (10p)
{
    return ay <= 1 ? ay : Tavşan(ay - 1) + Tavşan(ay - 2);
}
```

//3.soru

```
static bool Asal(int a) (5p)
{
    return Asal(a, a - 1);
}
static bool Asal(int a, int b) (5p)
{
    return b == 1 ? true : Kalan(a, b) == 0 ? false : Asal(a, b - 1);
}
static int Kalan(int a, int b) (5p)
{
    return a < b ? a : Kalan(a - b, b);
}
static int EnKüçük_AsalSayı(int a, int b) (10p)
{
    return Asal(a) ? a : EnKüçük_AsalSayı(a + 1, b);
}
```

//4.soru

```
class Ders (5p)
{
    public string Ad { get; set; } (5p)
    int _not;
    public int Not (5p)
    {
        get { return _not; }
        set { _not = value >= 1 && value <= 3 ? value : 1; }
    }
    public string HarfNot()
    {
        return Not == 1 ? "C" : Not == 2 ? "B" : "A"; (5p)
    }
    public override string ToString() (5p)
    {
        return String.Format("{0} : {1} = {2}", Ad, Not, HarfNot());
    }
}
```

## (2.ÖĞRETİM) NESNE TABANLI PROGRAMLAMA-1 FİNAL SINAVI SORULARI

### UYARILAR:

- \* 1,2,3,4.Soruların cevaplarında "**if, switch, for, while, do-while, foreach**" yazılmayacaktır.
- \* 1,2,3,4.Soruların cevaplarında '\*', '/', '%' işaretleri kullanılmayacaktır.
- \* Tüm metotlarda ve özelliklerin iç kısımlarında **sadece bir satır kod** yazılacaktır.
- \* 1,2,3.Sorularda sadece metotlar yazılacaktır.(Main metodu içerisi yazılmıştır)
- \* 4.Soruda sadece sınıf, metotlar ve özellikler yazılacaktır.(Main metodu içerisi yazılmıştır)
- \* 1,2,3,4.Sorularda metot ve özellik isimleri "Main" metodundan çağrıldığı gibi tanımlanacaktır.
- \* 1,2,3.Sorularda (4.soru hariç) metotlar **ÖZYİNELEMELİ (RECURSIVE)** olarak yazılacaktır.

**1.Soru:** Farklı miktarda sayının toplanmasını sağlayan metodu yazınız.

**Bilgi:** Dizi uzunluğunu elde etmek için "Length" özelliği kullanılır. (dizi.Length)

```
int adet = 0, toplam = Topla(_____ adet, 55, 88, 22, 77, 99, 11, 33, 66, 44);
```

```
Console.WriteLine("{0} adet sayının toplam:{1}", adet, toplam); //7 adet sayının toplamı:385
```

**2.Soru:** Aşağıdaki matematik sorusunun çözümünü yapan metotları oluşturunuz.

**Problem:** "Dört yanı duvarlarla çevrili bir yere **bir çift tavşan** konmuştur. Her çift tavşanın **bir ay** içinde **yen bir çift** tavşan yavruladığı, her yeni çiftin de erginleşmesi için **bir ay** gerektiği ve tavşanların ölmediği varsayılırsa, dört duvarın arasında **en az 100 çift** tavşanın olması için kaç ay geçmesi gerekir?"

**Çözüm:** Tavşan çiftlerinin sayısı aylara göre şu sıralamayı ortaya koymaktadır:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610,....

İlk iki sayı hariç, her sayı kendisinden önce gelen iki sayının toplamına eşittir.

```
int ay = Ay(100), tavşan = Tavşan(ay);
```

```
Console.WriteLine("Ay: {0}, Tavşan: {1} çift", ay, tavşan); //Ay:12, Tavşan: 144 çift
```

**3.Soru:** Aynı anda aynı yerden hareket eden iki yolcu otobüsünden birincisi 16 saatte, ikincisi 24 saatte seferini tamamlayıp geri dönüyor. Bu otobüsler kaç saat sonra tekrar karşılaşırlar?

**Bilgi:** Karşılaşma Süresi:  $(16 \times 24) / 8 = 48$  saat

```
Console.WriteLine("{0} saat sonra tekrar karşılaşırlar", KarşılaşmaSüresi(16, 24)); //48 saat
```

**4.Soru:** Aşağıda isimleri ve ortalamaları girilen öğrencilere göre sınıfı oluşturunuz.

\* Sınıf adı : **Öğrenci**

\* Özellikler: **Ad**(metin), **Ortalama**(çift, sadece 0 ile 4 arası değerler girilebilir)

\* Metotlar : **MezuniyetDerecesi** (Ortalama özelliğine göre 2'den küçükse ise "Mezun Değil", 2-2.99 arası ise "C", 3-3.49 arası ise B, 3.5-4 arası ise "A" sonucu döndürecek)

**ToString** (Ekran çıktısı "Ali : 2,80 = C" şeklinde olacak)

```
Console.WriteLine(new Öğrenci() { Ad = "Ali", Ortalama = 1.8 });
```

```
Console.WriteLine(new Öğrenci() { Ad = "Veli", Ortalama = 3.2 });
```

```
Console.WriteLine(new Öğrenci() { Ad = "Ayşe", Ortalama = 3.6 });
```

```
7 adet sayının toplamı : 385
```

```
Ay: 12, Tavşan: 144 çift
```

```
48 saat sonra tekrar karşılaşırlar
```

```
Ali : 2,80 = C
```

```
Veli : 3,20 = B
```

```
Ayşe : 3,60 = A
```

Tüm soruların ekran çıktısı:

Süre : 40 dakika

## (2.ÖĞRETİM) NESNE TABANLI PROGRAMLAMA-1 FİNAL SINAVI CEVAPLARI

//1.soru:

```
static int Topla(ref int adet, params int[] Sayılar) (10p)
{
    return adet == Sayılar.Length ? 0 : Sayılar[adet++] + Topla(ref adet, Sayılar); (15p)
}
```

//2.soru:

```
static int Ay(int adet) (5p)
{
    return Ay(0, adet);
}
static int Ay(int a, int b) (10p)
{
    return Tavşan(a) >= b ? a : Ay(a + 1, b);
}
static int Tavşan(int ay) (10p)
{
    return ay <= 1 ? ay : Tavşan(ay - 1) + Tavşan(ay - 2);
}
```

//3.soru

```
static int Çarpım(int a, int b) (5p)
{
    return b == 1 ? a : a + Çarpım(a, b - 1);
}
static int Bölüm(int a, int b) (5p)
{
    return a < b ? 0 : 1 + Bölüm(a - b, b);
}
static int Kalan(int a, int b) (5p)
{
    return a < b ? a : Kalan(a - b, b);
}
static int EBOB(int a, int b) (5p)
{
    return b == 0 ? a : EBOB(b, Kalan(a, b));
}
static int KarşılaşmaSüresi(int a, int b) (5p)
{
    return Bölüm(Çarpım(a, b), EBOB(a, b));
}
```

//4.soru

```
class Öğrenci(5p)
{
    public string Ad { get; set; }(5p)
    double _ortalama;
    public double Ortalama(5p)
    {
        get { return _ortalama; }
        set { _ortalama = value >= 0 && value <= 4 ? value : _ortalama; }
    }
    public string MezuniyetDerecesi()(5p)
    {
        return Ortalama < 2 ? "Mezun Değil" : Ortalama < 3 ? "C" : Ortalama < 3.5 ? "B" : "A";
    }
    public override string ToString()(5p)
    {
        return String.Format("{0,-5} : {1,3:0.00} = {2}", Ad, Ortalama, MezuniyetDerecesi());
    }
}
```