



BURSA ULUDAĞ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
2022 – 2023 Eğitim Öğretim Yılı Güz Yarıyılı
Ödev 2-1

Ferit Yiğit BALABAN

032190002

032190002@ogr.uludag.edu.tr

https://github.com/fybx/assignments/blob/main/BMB2009/Assignment2_1/Program.cs

Soru 1 - Aşağıdaki deseni çıkartan DispCrown isimli metodu yazınız. Metodun parametrik yapısı şöyle olmalıdır: public static void DispCrown(int n)
Buradaki n aşağıdaki desenlerdeki en yüksek sayıyı belirtmektedir.

```
1      1      1
12     212     21
123    32123    321
123443212344321
```

Burada n = 4'tür.

Cevap 1 - Kaynak Kod

```
public static class Assignment2_1
{
    public static void Main(string[] args)
    {
        string? input;
        while (true)
        {
            Console.WriteLine("n gir: ");
            input = Console.ReadLine();
            if (Convert.ToInt32(input) > 1)
            {
                DispCrown(Convert.ToInt32(input));
            }
        }
    }
}
```

```

public static void DispCrown(int n)
{
    int row = 1, j;
    for (; row < n + 1; row++)
    {
        for (j = 1; j < row + 1; j++)
            Console.Write(j);
        for (int ctr = 0; ctr < 2 * (n - row); ctr++)
            Console.Write(" ");
        for (j = row; j > 0; j--)
            Console.Write(j);
        for (j = 2; j ≤ row; j++)
            Console.Write(j);
        for (int ctr = 0; ctr < 2 * (n - row); ctr++)
            Console.Write(" ");
        for (j = row; j > 0; j--)
            Console.Write(j);
        Console.WriteLine("\n");
    }
}
}

```

Cevap 1 - Ekran Görüntüsü

```

~/r/assignments/B/Assignment2_1/b/D/net6.0 git main !1 > ./Assignment2_1
Soru seç (1 veya 2): 1
n gir:
2
1 1 1
1221221
n gir:
3
1 1 1
12 212 21
12332123321
n gir:
4
1 1 1
12 212 21
123 32123 321
123443212344321
n gir:
5
1 1 1
12 212 21
123 32123 321
1234 4321234 4321
1234554321234554321
n gir:
6
1 1 1
12 212 21
123 32123 321
1234 4321234 4321
12345 543212345 54321
12345665432123456654321
n gir:
9
1 1 1
12 212 21
123 32123 321
1234 4321234 4321
12345 543212345 54321
123456 65432123456 654321
1234567 7654321234567 7654321
12345678 876543212345678 87654321
12345678998765432123456789987654321

```

Cevap 1 – Açıklaması

En dışta yer alan ilk **for** döngüsü ile **row** değişkeni hangi satırda yer aldığımızı takip eder. Yani her bir satıra girilecek her karakter en dıştaki **for** döngüsünün her bir iterasyonunda belirlenir. İçeride bulunan 6 **for** döngüsü ise karakterlerin kurallı basımını sağlar.

1. for döngüsü en solda yer alan dik üçgenin çizilmesini sağlar.
2. for döngüsü sol orta boşluğu hesaplar ve ekrana bastırır. Bastırılacak boşluk sayısı ile hangi satırda olduğumuz arasında matematiksel bir ilişki vardır ve şöyle modellenir: $2 \cdot (n - row)$.
3. for döngüsü ortada yer alan tam üçgenin 1 dahil olacak biçimde sol parçasını çizer.
4. for döngüsü ortada yer alan tam üçgenin sağ parçasını çizer.
5. for döngüsü sağ orta boşluğu yukarıda verilen matematiksel ilişkiye göre hesaplanır.
6. for döngüsü sağa dayalı dik üçgeni çizer.

Tam bir iterasyon tamamlanmadan önce bir tane newline karakteri basılır ve döngü varsa bir sonraki satır için tekrar edilir.

Soru 2, 1. Kısım - Parametresi ile aldığı sayının Armstrong olup olmadığını ekrana yazdıran IsArmstrong isimli metodu yazınız. Metodun parametrik yapısı şöyledir: `public static bool IsArmstrong(int val)`

2. Kısım - Bu soruyu yaptıktan sonra 1'den n'e kadar olan tüm Armstrong sayıları

yazdıran metodu yazıp 1000000 değeriyle test ediniz. Metodun parametrik yapısı şöyle olmalıdır:

`public static void DispAllArmstrongs(int n)`

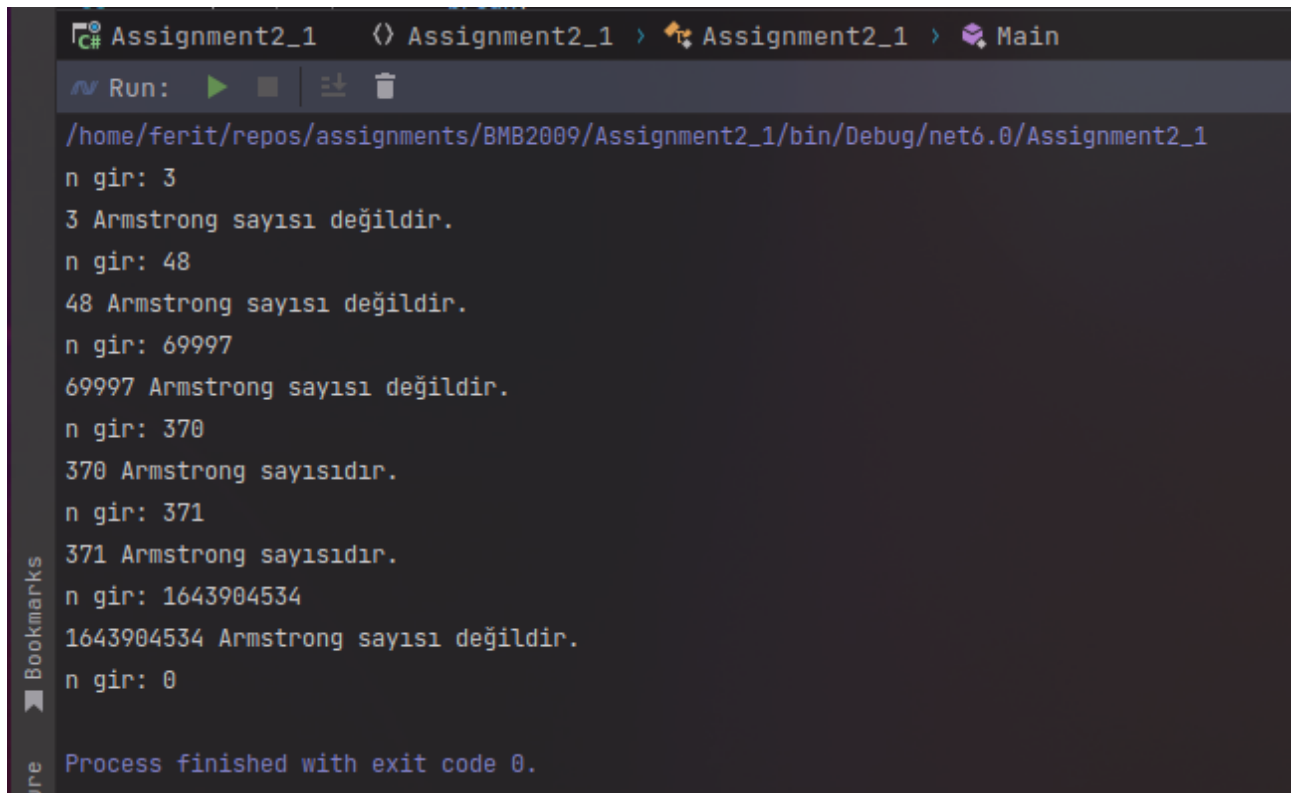
Cevap 2, 1. Kısım – Kaynak Kod

```
public static void Main(string[] args)
{
    int n;
    while (true)
    {
        Console.Write("n gir: ");
        n = Convert.ToInt32(Console.ReadLine()?.Trim());
        if (n is 0)
            break;
        Console.WriteLine("{0} {1}", n, IsArmstrong(n) ? "Armstrong sayısıdır." :
"Armstrong sayısı değildir.");
    }
}

public static bool IsArmstrong(int val)
{
    int n = val, digit, sum = 0;
    do
    {
        digit = n % 10;
        sum += digit * digit * digit;
        n /= 10;
    } while (n != 0);

    return val == sum;
}
```

Cevap 2, 1. Kısım – Ekran Görüntüsü



```
Assignment2_1 > Assignment2_1 > Assignment2_1 > Main
Run:
/home/ferit/repos/assignments/BMB2009/Assignment2_1/bin/Debug/net6.0/Assignment2_1
n gir: 3
3 Armstrong sayısı değildir.
n gir: 48
48 Armstrong sayısı değildir.
n gir: 69997
69997 Armstrong sayısı değildir.
n gir: 370
370 Armstrong sayısıdır.
n gir: 371
371 Armstrong sayısıdır.
n gir: 1643904534
1643904534 Armstrong sayısı değildir.
n gir: 0
Process finished with exit code 0.
```

Cevap 2, 1. Kısım – Açıklama

IsArmstrong metodu **val** parametresinin tuttuğu sayıyı üzerinde işlem yapılacak olan **n** lokal değişkenine aktarır. Anlık rakam değerinin tutmak için **digit**, toplam değerini tutmak için **sum** değişkenleri deklare edilir. **do-while** döngüsü içerisinde sağdan başlayarak her rakam alınır, bir iterasyon boyunca **digit** değişkeninde saklanır. **sum** değerine **digit**'in küpü eklenir. ($sum = sum + (digit)^3$). Her rakam için bu tekrarlanır ve döngü tamamlandığında eldeki **sum** ile baştaki **val** parametresinin değeri karşılaştırılır. Metottan geriye **true** dönerse *sayı Armstrong sayısıdır*, false dönerse değildir.

Cevap 2, 2. Kısım – Kaynak Kod

```
public static void Main(string[] args)
{
    Console.WriteLine("n gir: ");
    int n = Convert.ToInt32(Console.ReadLine()?.Trim());
    DispAllArmstrong(n);
}

public static void DispAllArmstrong(int n)
{
    for (int i = 1; i ≤ n; i++)
        Console.Write(IsArmstrong(i) ? $"{i} Armstrong sayısıdır.\n" : "");
}
```

Cevap 2, 2. Kısım – Ekran Görüntüsü

```
Assignment2_1  (<) Assignment2_1  >  Assignment2_1
Run:  [Run] [Stop] [Debug] [Close]
/home/ferit/repos/assignments/BMB2009/Assignment2_1/bin/Debug/net6.0/Assignment2_1
n gir: 1000000
1 Armstrong sayısıdır.
153 Armstrong sayısıdır.
370 Armstrong sayısıdır.
371 Armstrong sayısıdır.
407 Armstrong sayısıdır.

Process finished with exit code 0.
```

Cevap 2, 2. Kısım – Açıklama

DispAllArmstrong metodu **n** parametresine aldığı değere kadar olan sayıların her birini **IsArmstrong** metodunun **true** ya da **false** dönmesine bağlı olarak ekrana yazdırır. Dışta yer alan **for** döngüsü **i** değişkeninin 1'den n'e (n dahil) dönmesini sağlarken içeride yer alan **Console.Write** çağrısı ekrana ya boş bir string ya da Armstrong sayısı hakkında metin basar. Bu koşullandırmayı tek satıra indirmek için koşul ifadesi **IsArmstrong(n)** olan bir boolean ternary operatörü kullandım.

Soru Puanlaması

Cevap : 60

Ekran Görüntüsü : 15

Açıklama : 25