



**BURSA ULUDAĞ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**2022 – 2023 Eğitim Öğretim Yılı Güz Yarıyılı**  
**BMB2009 Ödev 3**

Ferit Yiğit BALABAN

032190002

[032190002@ogr.uludag.edu.tr](mailto:032190002@ogr.uludag.edu.tr)

**Soru 1** “1 - 2147483647 arasında girilen sayının metinsel karşılığını yazan programı kodlayınız.”

**Cevap 1** - Kaynak Kod

<https://github.com/fybx/assignments/blob/main/BMB2009/Assignment3/Program.cs>

```
namespace Assignment3;

public static class Assignment3
{
    private static void Main(string[] args)
    {
        bool @break = false;
        while (!@break)
        {
            Console.Write("Bir sayı girin: ");
            string? input = Console.ReadLine();
            @break = string.IsNullOrEmpty(input?.Replace(" ", ""));
            if (!@break)
            {
                int sayi = Convert.ToInt32(input);
                Console.WriteLine(sayi.0ku());
            }
        }
    }
}
```

```

public static class Extensions
{
    private static Dictionary<int, string> BasBir = new()
    {
        {0, ""}, {1, "bir"}, {2, "iki"}, {3, "üç"}, {4, "dört"}, {5, "beş"}, {6, "altı"}, {7, "yedi"}, {8,
"sekiz"}, {9, "dokuz"}
    };

    private static Dictionary<int, string> BasOn = new()
    {
        {0, ""}, {1, "on"}, {2, "yirmi"}, {3, "otuz"}, {4, "kırk"}, {5, "elli"}, {6, "altmış"}, {7, "yetmiş"}, {8,
"seksen"}, {9, "doksan"}
    };

    private static Dictionary<int, string> Binlik = new()
    {
        { 0, "" }, { 1, "bin" }, { 2, "milyon" }, { 3, "milyar" }
    };

    public static string Oku(this int deger)
    {
        int bin = 0, dig = 0, peek = 0, dpk = 0, tpk = 0, dgr = deger;
        string txt = "";
        do
        {
            int pos = 0;
            txt = bin == 0 || peek + dpk + tpk == 0 ? txt : $"{(Binlik[bin])} {txt.Trim()}";
            do
            {
                dig = dgr % 10;
                peek = (dgr / 10) % 10;
                dpk = (dgr / 100) % 10;
                tpk = (dgr / 1000) % 10;
                txt = pos switch
                {
                    0 => $"{(bin == 1 && dig == 1 && peek + dpk == 0 ? "" : BasBir[dig])} {txt}"
                        .Trim(),
                    1 => $"{BasOn[dig]} {txt}.Trim()",
                    _ => dig is 0 ? txt : $"{(dig is 1 ? "yüz" : BasBir[dig] + " yüz")} {txt}"
                };
                dgr /= 10;
                pos++;
            } while (pos < 3 && dgr != 0);
            bin++;
        } while (dgr != 0);
        return txt.Trim();
    }
}

```

## Cevap 1 - Ekran Görüntüsü

```

Run: /home/ferit/repos/assignments/BMB2009/Assignment3/bin/Debug/net6.0/Assignment3
Bir sayı girin: 1011
bin on bir
Bir sayı girin: 101000
yüz bir bin
Bir sayı girin: 2000100100
iki milyar yüz bin yüz
Bir sayı girin: 2147483647
iki milyar yüz kırk yedi milyon dört yüz seksen üç bin altı yüz kırk yedi
Bir sayı girin: 123
yüz yirmi üç
Bir sayı girin: 10
on
Bir sayı girin:

```

## Cevap 1 – Açıklaması

**Extensions** sınıfında birler ve onlar basamağındaki rakamların ismi için Dictionary veri tipini kullanarak bir sözlük lookup tablosu oluşturdum. **Binlik** ismindeki Dictionary ile ise sayıyı her üçlü grupta oluşan kısımlara göre isimlendirme için bir tablo kurdum.

İç içe iki **do-while** döngüsünden dıştaki döngü sayı 0 olana dek dönerken, içteki döngü *her üç rakamda bir* tamamlanmakta. Yani örnek olarak girilen 123,456,789 sayısı için içteki döngü toplam 3 defa çalıştırılacak. Bir kez 789 kısmını işleyerek “yedi yüz seksen dokuz” metni elde edilip kaydedilecek. İkinci döngü başlarken **bin** isimli değişkenin değeri “1” olacak. Bu değer *hangi üçlü grupta* bulunduğumuzu gösteriyor. **bin** değişkeninin değerinden dolayı *içteki döngü başlamadan hemen önce* “yedi yüz seksen dokuz” değerini tutan **metin** değişkeninin sol başına “bin” eklenecek. Burada başa “bin” yazısı ekleyen mekanizma karar verebilmek için ileriye *peek* ederek “bin” yazmanın doğru olup olmadığını hesaplıyor.

Her üçlü grup tamamlandıktan sonra “bin/milyon/milyar” yazısının getirilebilmesi için **peek + dpk + tpk** işlemi ile ileride rakam olup olmadığı değerlendiriliyor. Yani 123 sayısı için “bin yüz yirmi üç” dememiş oluyor. Aynı şekilde 234,567 sayısı için de “milyon iki yüz otuz dört bin beş yüz altmış yedi” denmiyor.

Buraya kadar özetlenen sistem, alınan sayıyı üçlü gruplara ayırdı, her grubu kendi içinde metne dönüştürdü ve sonunda da incelenen üçlü gruptan sonra solda başka bir grup varsa uygun prefix’i (bin, milyon, milyar) getirdi. Fakat bir üçlü grubu metne dönüştüren parçayı anlatmadım.

Üçlü grubun metne dönüştürülmesinde sağdan sola sıralı okunan rakamların pozisyonlanmasını **pos** değişkeni, takip eden 1., 2. ve 3. rakamı da sırasıyla **peek**, **dpk** ve **tpk** değişkenleri saklıyor. Bu işlemlerin hemen sonrasında bir **switch** ifadesi ile metnin yeni gösterimi hesaplanıyor. Burada **string interpolation** kullandım çünkü,

1. Kendi C# deneyimimde birkaç farklı amaç için parserlar, lexer yazmıştım
2. Daha kısa ve şık ifadelerle metin oluşturulabiliyor
3. Havalı

**switch** ifadesi **pos** değişkenine göre 3 farklı interpolasyon formatından birisini seçiyor, bu interpolasyon stringleri ise bazı noktalardan **ternary boolean** operatörü ile metne ekleme yapıp yapmamaya karar veriyor. Sırasıyla her **pos** değeri için çıkan format stringlerinin ne yaptığını anlatayım:

**0:** Eğer 2. üçlü grupta (yüz/on/binler basamağı) bulunuyorsak ve şu an baktığımız rakam 1 ise ve sıradaki 1. ve 2. sayılar 0 (yani o sayılar yok) ise şu anki rakamı (1) yazma, aksi takdirde rakamı yaz. Bu koşul ile 1000 sayısı “bir bin” şeklinde **DEĞİL** “bin” şeklinde yazılır.

**1:** Onlar basamağı için elimdeki rakam neyse direkt yazıyorum, eğer elimde 0 varsa **BasOn** lookup tablom boş bir string döndüğü için hiçbir şey yazmıyorum. 101 => “yüz \_ bir” = “yüz bir” oluyor.

**\_:** **pos** her 3’e eşit olduğunda sıfırlandığı için bu **default** durum aslında sadece **pos = 2** için kullanılmış oluyor. En kompleks karar ve formatlama burada yapılıyor. Eğer bu noktada elimizdeki rakam 0 ise direkt şimdiye dek oluşturulmuş metin döndürülüyor, **txt = txt** oluyor. Eğer elimizdeki rakam 1 ise metnin soluna sadece “yüz” ekleniyor, böylelikle 123 sayısı “bir yüz yirmi üç” değil “yüz yirmi üç” biçiminde yazılıyor. Eğer elimizdeki rakam 0 ve 1 dışında bir rakam ise **BasBir** lookup tablosundan alınan metin ile “yüz” birleşiyor.

### Soru Puanlaması

Cevap : 60

Ekran Görüntüsü : 15

Açıklama : 25