



BURSA ULUDAĞ ÜNİVERSİTESİ BİLGİSAYAR  
MÜHENDİSLİĞİ BÖLÜMÜ 2023 – 2024

Eğitim Öğretim Yılı Güz Yarıyılı

Görsel Programlama Dersi

Ödev 1

032190007-Adnan Topçu

### Çalışma Soruları 1 :

1. Şehirlerin ilçelerini tutan bir Hashtable nesnesi yaratınız. Sonra bir döngü içerisinde klavyeden şehir ismini isteyiniz, sözlükten ona karşılık gelen ilçeleri bulup yazdırınız. Açıklama: Hashtable nesnesi anahtar olarak şehir ismini, değer olarak da ilçe isimlerinin bulunduğu string dizisini tutmalıdır. Deneme işlemini birkaç şehir ve ilçe girerek yapabilirsiniz.
2. Bir string içerisindeki karakterlerin hepsinin farklı olup olmadığını aşağıda belirtilen yöntemle belirleyiniz. Sonra bir döngü içerisinde klavyeden string okuyarak sonuçları yazdırınız. Açıklama: Bir Hashtable nesnesi alıp stringh içerisindeki karakterleri tek tek bu collection'a ekleyiniz. Ancak eklemeyen önce de bu karakter collection'da var mı diye bakınız.
3. İki sözcüğün anagram olup olmadığını (yani tamamen aynı sayıda aynı karakterlerden oluşup oluşmadığını) belirleyen programı aşağıda açıklandığı gibi yazınız. Açıklama: Bir Hashtable ya da Dictionary nesnesi oluşturunuz. Bu sözlük nesnesinin anahtarı char türden değeri de int türden olsun. Değer ilgili karakterin sayısını belirtmektedir. Sonra birinci sözcüğün karakterlerini bu sözlükte arayınız, bulursanız onun sayacını bir artırınız. Bulamazsanız onu 1 sayaç değeriyle sözlüğe ekleyiniz. Sonra ikinci sözcüğün karakterlerini bu sözlükte arayıp sayacı bir eksiltiniz. Eğer karakter yoksa ya da onun sayaç değeri negatif olmuşsa bu iki sözcük anagram olamaz. Programı test etmek için bir döngü kullanınız. Döngüde iki sözcük kullanıcıdan istensin ve program bunların anagram olup olmadığını yazdırsın. Başarılar...

## 1.Problem:

### Kod:

```
using System;

using System.Collections;

using System.Collections.Specialized;

using System.Reflection.Metadata.Ecma335;

namespace adnan_tpc
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] Bursa = { "Nilüfer", "Gemlik", "Kestel","Mudanya","Osmangazi" };    // şehir bilgileri

            string[] Trabzon = { "Sürmene", "Araklı", "Of", "Arsin", "Yomra" };

            string[] Sakarya = { "Adapazarı", "Akyazı", "Ferizli", "Hendek", "Ferizli" };

            Hashtable hashtable = new Hashtable();    // hashtable nesnesi oluşturma

            hashtable.Add("bursa", Bursa);    // hashtable ekleme yapma s

            hashtable.Add("trabzon", Trabzon);

            hashtable.Add("sakarya", Sakarya);

            while (true)    // döngü içinde değer alma ve kontrol etme
            {
                Console.WriteLine("\nLütfen bir şehir ismi giriniz (Çıkmak için q basın)=> ");    //şehir ismi alma

                string k = Console.ReadLine();

                if (k == "q" || k == "Q") // q veya Q döngüden çıkar
                {
                    Console.WriteLine("Program sonlandırdı!!!"); //durum bilgisi

                    break;
                }

                else if (hashtable.ContainsKey(k.ToLower())) // girilen şehir ismi hashtableda var mı yok mu
                {
                    Console.WriteLine("İstenen şehir var=>>>"); // durum bilgisi

                    foreach (string s in (string[])hashtable[k.ToLower()]) { // şehrin ilçelerini yazdırma

                        Console.Write(s+" ");

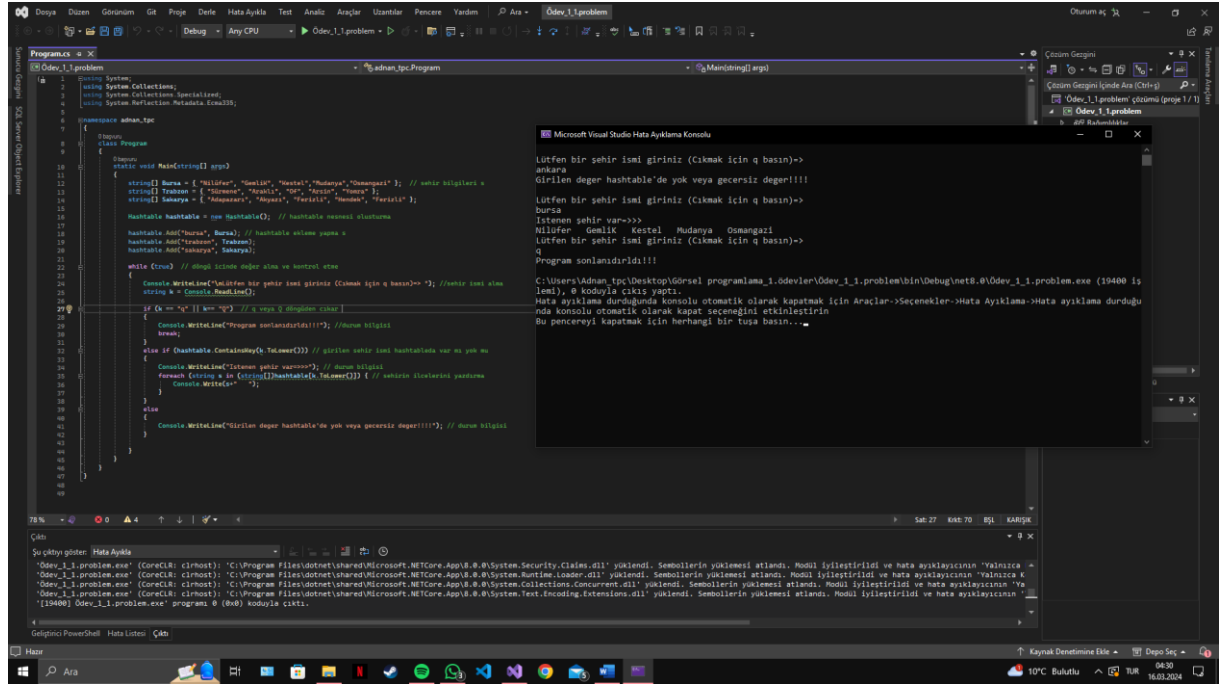
                    }
                }
                else
                {

```

```
Console.WriteLine("Girilen deger hashtable'de yok veya gecersiz deger!!!!"); // durum bilgisi
```

```
}  
}  
}  
}  
}
```

## Ekran görüntüsü:



## 2.Problem :

Kod:

```
using System;  
  
using System.Collections;  
  
using System.Globalization;  
  
using System.Reflection.Metadata.Ecma335;  
  
namespace adnan_tpc  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Girilen deger hashtable'de yok veya gecersiz deger!!!!"); // durum bilgisi  
        }  
    }  
}
```

```
bool flag = false; // bayrak bilgisi

Console.WriteLine("Lütfen bir string deger girin"); // string karakter alma

string k = Console.ReadLine().ToLower();

Hashtable hashtable = new Hashtable(); // hashtable nesnesi olusturma

foreach (char i in k) // girilen stringin karakter bilgisini kaydetme

{

    if (hashtable.ContainsKey(i))

    {

        flag = true;

        continue;

    }

    else

    {

        hashtable.Add(i, i);

    }

}

if (flag) // bayrak bilgisine göre bir stringin ayni veya farkli karakterlerden olustugunu yazdirma

{

    Console.WriteLine("Girilen string ayni karakter iceriyor"); // durum bilgisi

}

else

{

    Console.WriteLine("Girilen string farkli karakterlerden olusuyor"); //durum bilgisi

}

foreach (DictionaryEntry entry in hashtable) //hashtable yazdirma

{

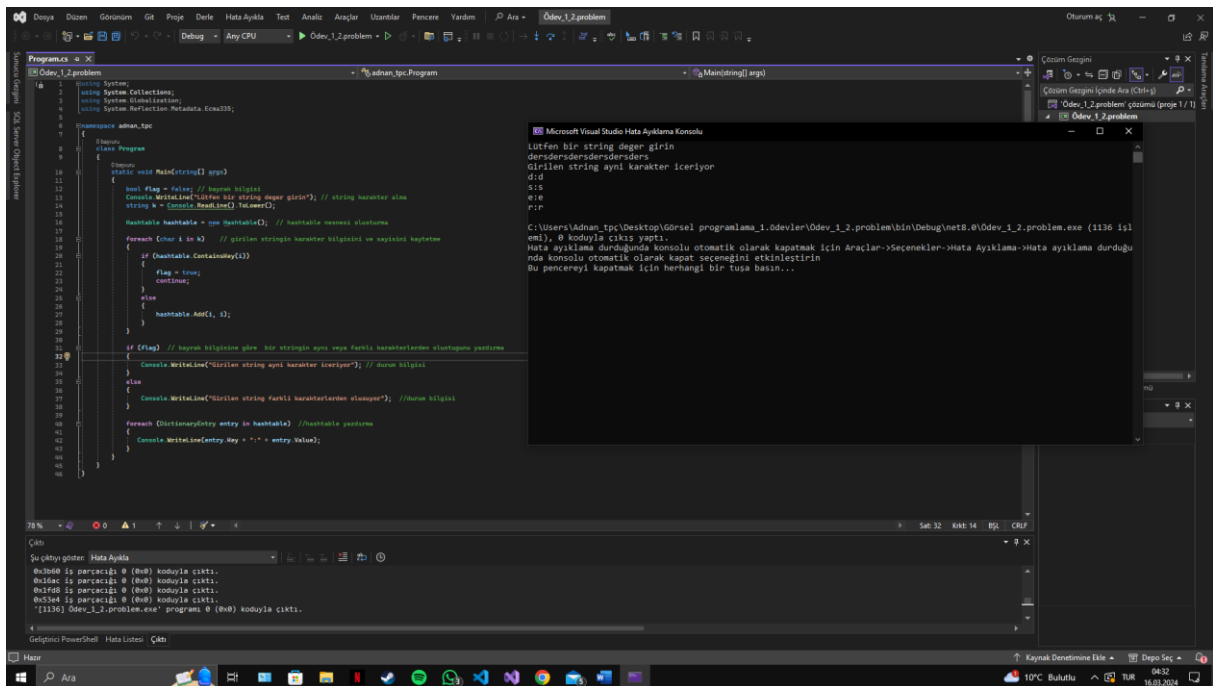
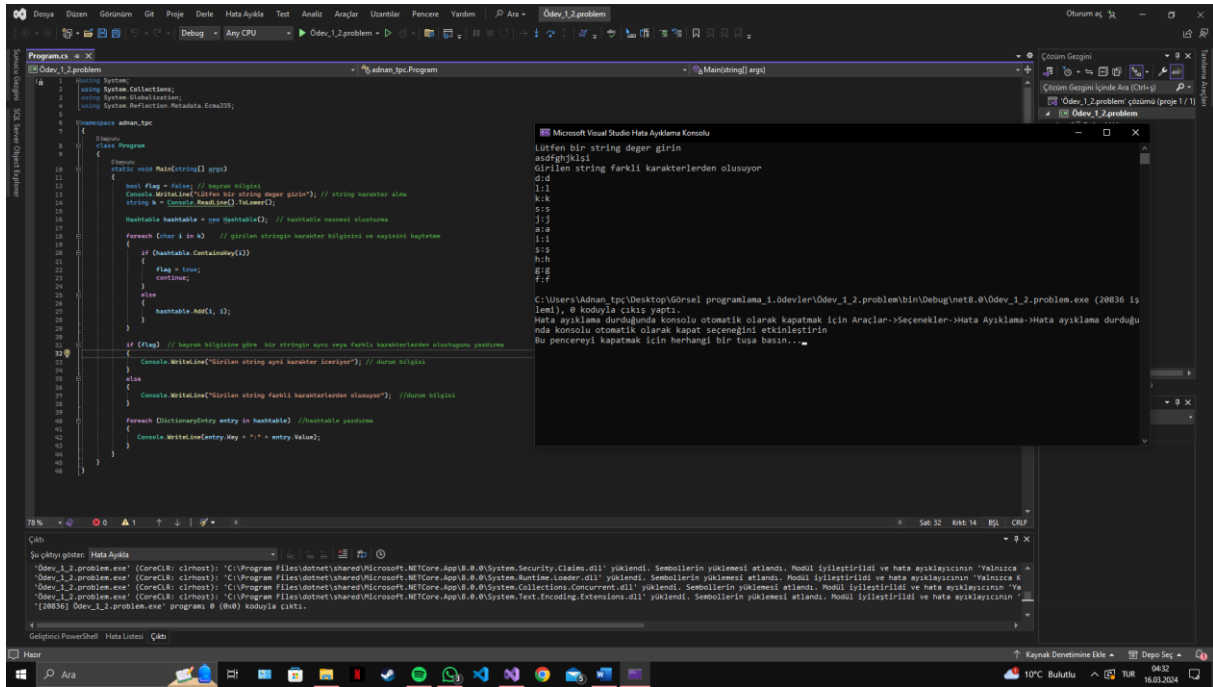
    Console.WriteLine(entry.Key + ":" + entry.Value);

}

}

}
```

Ekran görüntüsü:



### 3.Problem:

Kod:

```

using System;
using System.Collections;
using System.Globalization;
using System.Reflection.Metadata.Ecma335;

```

```
namespace adnan_tpc
{
    class Program
    {
        static void Main(string[] args)
        {
            bool flag = false; //bayrak bilgisi default olarak false ilerde anagram olup olmadığını kontrol etmek için

            Console.WriteLine("Lütfen 1. string degeri girin=>"); // string ifade almak

            string k = Console.ReadLine().ToLower();

            Console.WriteLine("Lütfen 2. string degeri girin=>"); // string ifade alma

            string l = Console.ReadLine().ToLower();

            Hashtable hashtable1 = new Hashtable(); //hashtable nesneleri olusturma

            Hashtable hashtable2 = new Hashtable(); //hashtable nesneleri olusturma

            foreach (char i in k)
            {
                if (hashtable1.ContainsKey(i)) //stringin karakterlerini ve karakter sayılarını hashtable yazdırma
                {
                    hashtable1[i]=(int)hashtable1[i]+1;

                    continue;
                }
                else
                {
                    hashtable1.Add(i, 1);
                }
            }

            foreach (char i in l) //stringin karakterlerini ve karakter sayılarını hashtable yazdırma
            {
                if (hashtable2.ContainsKey(i))
                {
                    hashtable2[i] = (int)hashtable2[i] + 1;

                    continue;
                }
                else
                {
                    hashtable2.Add(i, 1);
                }
            }
        }
    }
}
```

```

    }

    //2 hashtable bilgisini karsilastirma

    foreach (char i in hashtable1.Keys) { //1. hashtableda gezme

        if (hashtable2.ContainsKey(i)) // 1. hashtable key degerlerini 2. hashtable da var mı yok mu kontrol etme

        {

            if ((int)hashtable2[i] != (int)hashtable1[i]) // varsa bu karakterin 2 hashtableda sayılarını kontrol ediyorum

            {

                flag = true; // degerler farklıysa bayragı true yapıyorum

                break;

            }

        }

        else // yoksa bayrak true yapıyorum

        {

            flag = true;

            break;

        }

    }

    if (flag) // bayraga gore durum yazdırma

    {

        Console.WriteLine("Girilen stringler anagram degildir"); //durum bilgisi

    }

    else

    {

        Console.WriteLine("Girilen stringler anagramdir"); // durum bilgisi

    }

    foreach (DictionaryEntry entry in hashtable1) //1. hashtable yazdırma

    {

        Console.WriteLine(entry.Key + ":" + entry.Value);

    }

    Console.WriteLine("#####");

    foreach (DictionaryEntry entry in hashtable2) //2. hashtable yazdırma

    {

        Console.WriteLine(entry.Key + ":" + entry.Value);

    }

}

}

```



## Ekran görüntüsü:

