



BURSA ULUDAĞ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
2023-2024 EĞİTİM ÖĞRETİM YILI BAHAR DÖNEMİ
GÖRSEL PROGRAMLAMA ÖDEV

MURAT BERK YETİŞTİRİR

032290008

032290008@ogr.uludag.edu.tr

Soru 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.

Soru 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 string içerisindeki karakterleri tek tek bu collection'a ekleyiniz. Ancak eklemeyen önce de bu karakter collection'da var mı diye bakınız.

Soru 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<K, V> 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.

CEVAP KODU:

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Homework1
{
    internal class Program
    {
        static bool IsAnagram(string str1, string str2)
        {
            Hashtable ht2 = new Hashtable();
            if (str1.Length != str2.Length)
                return false;

            foreach (char s in str1)
            {
                if (ht2.ContainsKey(s))
                    ht2[s] = (int)ht2[s] + 1;
                else
                    ht2[s] = 1;
            }
            foreach (char s in str2)
            {
                if (ht2.ContainsKey(s))
                {
                    ht2[s] = (int)ht2[s] - 1;
                    if ((int)ht2[s] < 0)
                        return false;
                }
                else
                    return false;
            }
            foreach (int count in ht2.Values)
            {
                if (count != 0)
                    return false;
            }
            return true;
        }
    }
}
```

```

static bool IsUnique(string str)
{
    Hashtable ht1 = new Hashtable();
    foreach (char c in str)
    {
        char lowercaseChar = char.ToLower(c); // Küçük harfe çevir
        if (ht1.ContainsKey(lowercaseChar))
        {
            return false;
        }
        ht1.Add(lowercaseChar, true);
    }
    return true;
}

static void Main(string[] args)
{
    while (true)
    {
        Console.Write("Bir string girin (çıkamak için 'q' tuşuna basın): ");
        string inputString = Console.ReadLine();

        if (inputString.ToLower() == "q")
            break;

        if (IsUnique(inputString))
        {
            Console.WriteLine("Tüm karakterler farklıdır.");
        }
        else
        {
            Console.WriteLine("Aynı karakterler bulunmaktadır.");
        }
    }
    Console.WriteLine("\n-----\n");
    while (true)
    {
        Console.Write("İlk kelimeyi girin (çıkamak için 'q' tuşuna basın): ");
        string word1 = Console.ReadLine();
        if (word1.ToLower() == "q")
            break;

        Console.Write("İkinci kelimeyi girin: ");
        string word2 = Console.ReadLine();

        if (IsAnagram(word1.ToLower(), word2.ToLower()))
            Console.WriteLine("İki kelime anagramdır.");
        else
            Console.WriteLine("İki kelime anagram değildir.");
    }
    Console.WriteLine("\n-----\n");

    Hashtable ilceler = new Hashtable();
    ilceler.Add("Balıkesir", new string[] { "Altıeylül", "Ayvalık", "Balya",
"Bandırma", "Bigadiç", "Burhaniye", "Dursunbey", "Edremit", "Erdek" });
    ilceler.Add("Bilecik", new string[] { "Bozüyük", "Gölpazarı", "İnhisar", "Merkez",
"Osmaneli", "Pazaryeri", "Söğüt", "Yenipazar" });
    ilceler.Add("Bursa", new string[] { "Büyükorhan", "Gemlik", "Gürsu", "Harmancık",
"İnegöl", "İznik", "Karacabey", "Keles", "Kestel", "Mudanya", "Mustafakemalpaşa", "Nilüfer",
"Orhaneli", "Orhangazi", "Osmangazi", "Yenişehir", "Yıldırım" });
    ilceler.Add("Çanakkale", new string[] { "Ayvacık", "Bayramiç", "Biga", "Bozcaada",
"Çan", "Eceabat", "Ezine", "Gelibolu", "Gökçeada", "Lapseki", "Merkez", "Yenice" });
    ilceler.Add("Edirne", new string[] { "Enez", "Havsa", "İpsala", "Keşan",
"Lalapaşa", "Merkez", "Meriç", "Süloğlu", "Uzunköprü" });
    ilceler.Add("İstanbul", new string[] { "Adalar", "Arnavutköy", "Ataşehir",
"Avcılar", "Bağcılar", "Bahçelievler", "Bakırköy", "Başakşehir", "Bayrampaşa", "Beşiktaş",
"Beykoz", "Beylikdüzü", "Beyoğlu", "Büyükkçekmece", "Çatalca", "Çekmeköy", "Esenler",
"Esenyurt", "Eyüpsultan", "Fatih", "Gaziosmanpaşa", "Güngören", "Kadıköy", "Kağıthane",
"Kartal", "Küçükçekmece", "Maltepe", "Pendik", "Sancaktepe", "Sarıyer", "Silivri",

```

```

"Sultanbeyli", "Sultangazi", "Şile", "Şişli", "Tuzla", "Ümraniye", "Üsküdar", "Zeytinburnu"
});
    ilceler.Add("Kırklareli", new string[] { "Babaeski", "Demirköy", "Kofçaz",
"Lüleburgaz", "Merkez", "Pehlivan köyü", "Pınarhisar", "Vize" });
    ilceler.Add("Kocaeli", new string[] { "Başiskele", "Çayırova", "Darıca",
"Derince", "Dilovası", "Gebze", "Gölcük", "İzmit", "Kandıra", "Karamürsel", "Kartepe",
"Körfez" });
    ilceler.Add("Sakarya", new string[] { "Adapazarı", "Akyazı", "Arifiye", "Erenler",
"Ferizli", "Geyve", "Hendek", "Karapürçek", "Karasu", "Kaynarca", "Kocaali", "Pamukova",
"Sapanca", "Serdivan", "Söğüt", "Taraklı" });
    ilceler.Add("Tekirdağ", new string[] { "Çerkezköy", "Çorlu", "Ergene",
"Hayrabolu", "Kapaklı", "Malkara", "Marmaraereğlisi", "Muratlı", "Saray", "Süleymanpaşa",
"Şarköy" });
    ilceler.Add("Yalova", new string[] { "Altınova", "Armutlu", "Çınarcık",
"Çiftlikköy", "Merkez", "Termal" });

    Console.WriteLine("Lütfen bir şehir ismi giriniz: ");
    string cityName = Console.ReadLine();

    if (ilceler.ContainsKey(cityName))
    {
        Console.WriteLine($"{cityName} şehrinin ilçeleri:");
        string[] districts = (string[])ilceler[cityName];
        foreach (var district in districts)
        {
            Console.WriteLine(district);
        }
    }
    else
    {
        Console.WriteLine("Girdiğiniz şehir bulunamadı.");
    }
    Console.ReadKey();
}
}
}

```

CEVAP EKRAN GÖRÜNTÜSÜ:

```

C:\Users\Murat Berk\Desktop
Bir string girin (çıkmaq için 'q' tuşuna basın): Murat
Tüm karakterler farklıdır.
Bir string girin (çıkmaq için 'q' tuşuna basın): Berk
Tüm karakterler farklıdır.
Bir string girin (çıkmaq için 'q' tuşuna basın): q

-----

İlk kelimeyi girin (çıkmaq için 'q' tuşuna basın): Murat
İkinci kelimeyi girin: Berk
İki kelime anagram değildir.
İlk kelimeyi girin (çıkmaq için 'q' tuşuna basın): q

-----

Lütfen bir şehir ismi giriniz: Bursa
Bursa şehrinin ilçeleri:
Büyükorhan
Gemlik
Görsu
Harmanlık
İnegöl
İznik
Karacabey
Keles
Kestel
Mudanya
Mustafakemalpaşa
Nilüfer
Orhaneli
Orhangazi
Osmangazi
Yenişehir
Yıldırım

```