

**EGE UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

**204 DATA STRUCTURES (3+1)**

**2024–2025 FALL SEMESTER**

**PROJECT-3 REPORT**

**ARAMA AĞAÇLARI, YIĞINLAR ve HASH TABLOSU**

**DELIVERY DATE**

26/12/2024

**PREPARED BY**

Taha Yusuf Şentürk

İçindekiler

[1) EgeDeniziB\_Ağacı 4](#_Toc184929056)

[1.a EgeDeniziB Ağacının Oluşturulması ve Sıralı Olarak Ekrana Listelenmesi 4](#_Toc184929057)

[1.a.1 Kodlar 4](#_Toc184929058)

[1.a.2 Açıklama 4](#_Toc184929059)

[1.b Derinlik Bulma, Ağacı Listeleme, Düğüm Sayısı Buldurma, Dengeli Ağaç Derinliği Hesaplama 4](#_Toc184929060)

[1.b.1 Kodlar 4](#_Toc184929061)

[1.b.2 Ekran görüntüleri 4](#_Toc184929062)

[1.b.3 Açıklama 4](#_Toc184929063)

[1.c Arama Listeleme 4](#_Toc184929064)

[1.c.1 Kodlar 4](#_Toc184929065)

[1.c.2 Ekran görüntüleri 4](#_Toc184929066)

[1.d Özyineli olarak Dengeli Bir Ağaç Oluşturulması 4](#_Toc184929067)

[1.d.1 Kodlar 4](#_Toc184929068)

[1.d.2 Ekran görüntüleri 4](#_Toc184929069)

[1.d.3 Açıklama 5](#_Toc184929070)

[2) HASH TABLE 5](#_Toc184929071)

[2.a EgeDeniziB Nesnelerini Hash Table’a Yerleştirme 5](#_Toc184929072)

[2.a.1 Kaynak Kod 5](#_Toc184929073)

[2.a.2 Ekran Görüntüleri 5](#_Toc184929074)

[2.b Hash Table Güncelleme 5](#_Toc184929075)

[2.b.1 Kaynak Kod 5](#_Toc184929076)

[2.b.2 Ekran Görüntüsü 5](#_Toc184929077)

[2.b.3 Açıklama 5](#_Toc184929078)

[3) HEAP VERİ YAPISI (SINIFI) 5](#_Toc184929079)

[3.a Heap 5](#_Toc184929080)

[3.a.1 Kaynak Kod 5](#_Toc184929081)

[3.b Max Heap’e Yerleştirme 5](#_Toc184929082)

[3.b.1 Kaynak Kod 5](#_Toc184929083)

[3.c Max Heap Listeleme 5](#_Toc184929084)

[3.c.1 Kaynak Kod 5](#_Toc184929085)

[3.c.2 Ekran Görüntüsü 6](#_Toc184929086)

[3.c.3 Açıklama 6](#_Toc184929087)

[4) SIRALAMA 6](#_Toc184929088)

[4.a Sıralama Algoritmaları 6](#_Toc184929089)

[4.a.1 Kaynak Kod 6](#_Toc184929090)

[4.a.2 Açıklama 6](#_Toc184929091)

[4.b Zaman Karmaşıklığının Hesaplanması 6](#_Toc184929092)

[4.b.1 Açıklama 6](#_Toc184929093)

[4.c Çalışma Zamanının Hesaplanması 6](#_Toc184929094)

[4.c.1 Kaynak Kod 6](#_Toc184929095)

[4.c.2 Açıklama 6](#_Toc184929096)

[5) Görselleştirme 6](#_Toc184929097)

[5.a Görselleştirmenin Etkisi 6](#_Toc184929098)

[5.a.1 Açıklama 6](#_Toc184929099)

[5.b Karşılaştırma Yapılması 6](#_Toc184929100)

[5.b.1 Açıklama 6](#_Toc184929101)

[6) Özdeğerlendirme Tablosu 7](#_Toc184929102)

# 1) EgeDeniziB\_Ağacı

//The platform, version, and programming language used

.NET, 4.7.1,C#(windows11)

## 1.a EgeDeniziB Ağacının Oluşturulması ve Sıralı Olarak Ekrana Listelenmesi

### 1.a.1 Kodlar

|  |
| --- |
| internal class EgeDeniziB  {  public string balıkAdı;  public string bilgi;  public string BalıkAdı { get; set; }  public Tree kelimelerAğacı;  public Tree KelimelerAğacı { get; set; }    public EgeDeniziB() {    }  public EgeDeniziB(string name, string words)  {//words'de balık hakkındaki bütün bilgiler tek paragraf olarak alınır, parçalara bölünüp döngülerle kelimelerAğacı'na eklenir.  this.balıkAdı = name;  Tree örnekAğaç = new Tree();//ağacı direk setleyemedim(getter - setterda sorun yaşadım) onun yerine örnek bir ağaç oluşturup onu sınıfın ağacına bağladım/setledim.(javadaki this mantığıyla ilerledim.)  string[] asılKelimeler=divideWords(words);  this.bilgi=words;    foreach (var item in asılKelimeler)//paragraftaki asıl kelimeleri ağaca ekliyoruz  {  örnekAğaç.root=örnekAğaç.insert(örnekAğaç.root, item);  //KelimelerAğacı.root=KelimelerAğacı.insert(KelimelerAğacı.root, item);  //this.kelimelerAğacı.root=this.kelimelerAğacı.insert(this.kelimelerAğacı.root,item); //bunları denedim olmadı    }  this.kelimelerAğacı = örnekAğaç;    }  public string[] divideWords(string y) {  string[] z = y.Split(new char[] { ' ', ',', '.', ';', ':','(',')','-','0','1','2','3','4','5','6','7','8','9' });//paragrafta kelime olmayan yerleri ayrıştırıyoruz  string[] realWrods;//yukardaki split metodunda fazladan null("") durumları ortaya çıktı null harici asıl kelimeleri buraya atacağım.  int count = 0;  for (int i = 0; i < z.Length; i++)//asıl kelime sayısını buluyoruz.  {  if (z[i] != "" )  {  count++;  }  }  realWrods = new string[count];//initialize işlemi tamamlandı  int sayaç = 0;//realWords'de dolaşmak için bir tane daha sayaç attım  for (int i = 0; i < z.Length; i++)//bu noktada count elde edildi ve asıl kelimeler için kontrol işlemi başladı  {  if (z[i] != "")  {  realWrods[sayaç] = z[i];  sayaç++;  }  }  return realWrods;  }  }  EgeDeniziB balık1= new EgeDeniziB("Akya Balığı", "Carangida familyasının bir üyesi olan akya balığı, boyutları ve zevkli avcılığı ile olta balıkçılarının avlanırken en çok heyecanlandığı deniz balıklarından biridir. Akya olarak bilinen bu tür, aynı zamanda liça balığı olarak da adlandırılır.Maksimum yetişkin boyutları, 200 cm uzunluk ve 50 kg ağırlıktır. Ancak, genellikle Ege denizinde bulunanların 120 cm civarına kadar büyüdükleri görülmüştür.Üreme dönemleri olan Mayıs ve Ağustos arasında yumurtlamak için gittikleri nehir birleşimlerinde kefaller ve sardalyaları yiyerek güç toplarlar. Ana besinleri kefal yavrularıdır.Nehir ağızlarına yakın alanlarda daha küçük boyuttakiler bulunurken büyük boyutlu akyalar, 40-50 metre derinlikte ve kıyıya uzak olmayan alanlarda yaşar.");  EgeDeniziB balık2=new EgeDeniziB("Yaygın Aslan Balığı", "Hint-Pasifik bölgesine özgü bir aslan balığı varyantı olan bu tür, bilimsel olarak Pterois miles olarak adlandırılmıştır. Benzerlikleri dolayısıyla kırmızı aslan balığı ile karışıtırılırlar.Hint-Pasifik bölgesinin yanı sıra ülkemiz sularından Ege denizinde de yaşadıkları biliniyor. Genel aslan balığı çeşitleri gibi bu tür de zehirli bir türdür.Sırtında toplamda 13 adet diken bulunur. Renkleri kırmızı, gri veya ten rengi tonlarındadır. Geceleri avlandıklarından gündüzleri pek aktif bir tür değillerdir. Küçük kabuklular ve balıklar ile beslenirler.Yetişkin boyutları 35 cm’ye kadar ulaşabilir. Ege balıkları arasında olsalar da yaygın olarak Kızıldeniz’de ve Endenozya sularında görülürler.");  EgeDeniziB balık3=new EgeDeniziB("Barbunya", "Gerçek adı Barbunya balığı olsa da halk arasında Barbun adı ile bilinir. Mullidae familyasından olan bu türe Mullus barbatus bilimsel adı verilmiştir.Ege denizi gibi sıcak ve ılıman suların kıyıya yakın kumlu ve çamurlu bölgelerinde yaşarlar. Nadiren de olsa kayalık alanlarda görülmektedirler. Tekir balığına benzerlikleri ile bilinirler. Aradaki farkı anlamak için burun, göz altı ve sırt yüzgecine bakılmalıdır.Maksimum boyutları 40 cm’dir ancak, Ege Denizi bölgesindekiler genellikle 18-20 cm civarında olurlar. Ege balıkları arasında ticari öneme sahip, eti lezzetli balıklardandır.");  EgeDeniziB balık4=new EgeDeniziB("Çipura", "Çipura balığı diğer adıyla Çupra, Ege denizi balık türleri arasında yer alsa da aslında Akdeniz bölgesinde de yaşarlar.Ilıman sulara sahip bölgelerin, kumlu ve çamurlu bölgelerinde yaşarlar. Zaman zaman nehir ağızları ve lagünlerde bulundukları da olur.200 gr ve üzeri olanlar Çipura, bunun altında olanlar ise Lidaki olarak isimlendirilir. Kuvvetli bir çeneye sahiptir ve bu sayede kabukluları kolayca yiyebilir.Avlanmak isteniyorsa yaz aylarında kıyıya yakın, kış aylarında 30-40 metre derinlikte avlanabilirler. Kışın boyut olarak daha iri Çipuralar daha derin noktalara inerler. Ticari değeri yüksek ve eti oldukça lezzetli ege denizi balıklarındandır.");  EgeDeniziB balık5 = new EgeDeniziB("Çitari", "Görüntü olarak Çipura balığına çok benzerdir. Halk arasında Sarpa balığı olarak bilinirler. Yüzeyden 70 metre derinliğe kadar uzanan yaşam alanları vardır.Maviye çalan gri renkli üst kısımları, gümüş renkli yan kısımları vardır. Maksimum olarak 51 cm uzunluk ölçülmüş olup, genellikle 15-30 cm civarında görülürler.Ege denizi balık türleri arasında ticari değeri yoktur. Yapılan araştırmalarda tam olarak tespit edilemese de balığın tükettiği bir alg neticesinde, sarpa yiyen insanlarda halüsinojenik balık sarhoşluğu denilen rahatsızlığa sebep olur.");  EgeDeniziB balık6 = new EgeDeniziB("Çizgili Hani Balığı", "Hani balığı, Orfoz’un da içinde bulunduğu Serranidae familyasından ve Levreğin içinde bulunduğu Serraninae alt türünden bir balıktır. Bilimsel adı Serranus scriba’dır. Çizgili hani balığı Ege denizi balıkları arasında yer almaktadır ancak hani balıkları genel olarak Akdeniz ve Karadeniz’de yaygındırar.Ülkemiz denizlerinden Ege Denizi’ne özgü Çizgili hani, 5 ila 150 metre arasında yaşam alanına sahiptir. Gündüzleri kayalıkların oluşturduğu mağaralarda geçirirken, alacakaranlık ve geceleri avlanmak için ortaya çıkar.Genellikle yalnız yaşadıkları görülse de küçük sürüler halinde yüzdüklerine de rastlanmaktadır. Uzunlukları en fazla 25 cm’ye ulaşır. Kabuklular, kafadanbacaklılar ve küçük balıklar ile beslenir. Ticari değeri olmayan, eti lezzetli Ege balıklarından biridir.");  EgeDeniziB balık7 = new EgeDeniziB("Dikenli vatoz", "Dikenli vatoz, Dasyatidae familyasından bir vatoz balığı türüdür. Akdeniz ve Karadeniz bölgelerimizde de görülen bu tür, daha fazla görülmesi sebebiyle Ege Denizi balık türleri arasında bulunmalıdır.Çok derin sularda değil, genellikle 60 metre civarı derinlikte ve çamurlu alanlarda yaşarlar. Ana besinleri dipte yaşayan kabuklu türleridir. Ek olarak yumuşakçalar, solucanlar ve küçük balıklarla da beslenir.Ege denizinde bulunan dikenli vatozlar, genellikle 40-45 cm civarında veya daha küçük boyuttadır. Tırtırlı kuyruk ve iğneleri yüzünden insanlar için tehlikeli balık türleri arasındadır. Ticari değeri yoktur.");  EgeDeniziB balık8=new EgeDeniziB("Eşkina", "Eşina balığı, Sciaenidae familyasından bir deniz balığı türüdür. Bilimsel adı Sciaena umbra olan bu tür, Akdeniz ve Ege Denizi bölgelerimizde bulunmaktadır. Eşkina balığı amatör avcılık yapanların sıklıkla karşılaştığı bir türdür.Yavru eşkinalar kıyı şeritlere yakın nehir ağızlarında yaşarken, yetişkinler 5 metre ile 200 metre arasında yaşarlar. Ege balıkları listemize dahil ettiğimiz eşkina, en fazla 60 cm boyuta ulaşabilir ancak, ülkemiz sularında genellikle 30-40 cm civarında görülürler.Bu balığı sıradışı kılan bir özelliği de vardır. Balığın kafasından, alın bölgesinden 1 cm çapında taş çıkar. Bilimsel olarak kanıtlanmamış olsa da bu taş limon ile eritilerek tüketildiğinde böbrek taşı rahatsızlığına iyi geldiği düşünülür.Ticari değer açısından orta seviyelerdedir. Genellikle balık lokanlarında servis edilen, lezzetli bir türdür.");  EgeDeniziB balık9=new EgeDeniziB("Horozbina Balığı", "Horozbina, Blenniidae familyasından 900 civarı alt türe sahip, hem tuzlu hem de tatlı sularda (küçük bir bölümü) yaşayan alt varyantlara sahip balık türlerinin ortak adıdır.Horozbina tür adı altında 150 farklı cins ve 900 civarı da varyant tanımlanmıştır. Tatlı su blennysi olarak bilinen bir türü, nadiren de olsa akvaryumlarda beslenmektedir.Genellikle küçük boyutlara sahip olabilen bu balığın, yılan balığına benzer varyantlarıyla 55 cm boyutuna ulaştığı görülmüştür. İri bir göz ve ağız yapısına sahiptir.Zamanlarının büyük kısmını deniz tabanında ve kayalık yarıklarında geçirirler. Ticari değeri olmayan, Akdeniz ve Ege Denizi balıklarından biridir.");  EgeDeniziB balık10=new EgeDeniziB("İskaroz ", "Genellikle ılıman ve sıcak sulara sahip denizlerde yaşarlar. Bilimsel adı Sparisoma cretense olan tür Scaridae familyasındandır.Parlak renklere sahip bir balıktır. Görünüşü itibariyle sazan balığının tropik halini andırır. Papağan isminin verilmesi ağızlarının papağan gagasına benzemesindendir.Erkekler daha koyu tondaki renklere sahipken dişiler, kırmızı-turuncu tonlarındadır. Zamanlarının çoğu mercan resiflerinde geçer. Burada deniz yosunu ile beslenirler.Yetişkin dönemlerinde en fazla 50 cm olabilirler ancak, ülkemizdeki iskaroz balıkları 20-30 cm civarındadır. Ticari değeri bulunmayan iskarozun eti orta lezzettedir.");  EgeDeniziB balık11 = new EgeDeniziB("İskorpit Balığı", "İskorpit, Ege balıkları arasında zehirli türlerden biridir. Yüzgeçlerinde bulunan dikenlere dokunulduğunda kişiyi zehirler ve bölgede kızarıklık, şişlik görülür.Bu etki 2-3 gün devam eder. Amonyak kullanılarak tedavi edilebilir. Özellikle amatör balıkçılar tarafından kırlangıç balığı ile karıştırılıp yaralanmalara sebep olmaktadır. Tutulduğunda dikenlerine dokunmamaya dikkat edilmelidir.Kayıt edilmiş maksimum uzunluk 37 cm iken, ülkemiz sularında yetişkin olanları 20-25 cm civarında görülürler. Akdeniz, Karadeniz ve Ege denizi balık türlerinden biridir.İskorpit, zehirli balıklardan biri olsa da eti lezzetli olan ve barındırdığı vitamin ve yağlar ile insan sağlığına çok yararlı bir balıktır.");  EgeDeniziB balık12 = new EgeDeniziB("İsparoz", "Sparidae familyasından olan isparinin bilimsel adı Diplodus annularis’dir. Bu balık genellikle ılıman sahil bölgelerine yakın alanlarda yaşar. Ege balıklarından biridir ve diğer denizlerimizde de bulunur.Maksimum 25 cm uzunluğa erişebilirler. Ülkemiz sularında 15-18 cm civarında ispariler bulunur. Sportif balıkçılıkta sevilen bir balık türüdür. Genç olanları kışın lagünlere giderek beslenirler. Karides, yavru balıklar ve kurtçuklar ana besinleridir.Ticari değeri düşüktür ve genellikle olta balıkçılığı ile avlanırlar. Eti lezzetli ve az kılçıklı bir balıktır.");  EgeDeniziB balık13 = new EgeDeniziB("İzmarit Balığı", "İzmarit balığı, Ege denizi balık türleri arasında yer alıyor ancak Akdeniz bölgelemizde daha yaygın bulunur. Bilimsel adı Spicara smaris olan izmarit, Sparidae familyasından bir deniz balığı türüdür.Ilık sulara sahip bölgelerin kayalık, çamurlu dip kısımlarında yaşamaktadır. En fazla 25 cm boyutuna (erkekler) ulaşan bu tür, ülkemizde genellikle 15 cm civarında görülür.Ülkemiz sularında iki tür izmarit yaşamaktadır; İstargilos ve Menekşe izmarit. Eti lezzetli bir balıktır ve genellikle önce pişirilip sonra ayıklanır.");  EgeDeniziB balık14=new EgeDeniziB("Kalkan Balığı", "Kalkan balığı (Scophthalmus maximus), gözleri vücudunun solunda olan, Scophthalmidae familyasından bir deniz balığı türüdür. Vücudunun sağını deniz tabanına yatmak için kullanır.Ülkemizin tüm denizlerinde yaşayan bir türdür. Yaşam alanları 20 metre ile 70 metre arasındadır. Tipik balık türlerinden farklı olarak yuvarlak bir vücuda sahiptir. Maksimum 1 metre uzunluğa erişebilirler. Ülkemiz denizlerinde ise yetişkinler 60-70 cm civarında görülür.Henüz yavruyken gözleri sağda ve solda ayrı durmaktadır. 8-10 cm civarına ulaştıklarında sağ göz vücutlarının sol tarafına kaymaya başlar.Ticari değeri yüksek, eti çok lezzetli ege balıkları arasındadır.");  EgeDeniziB balık15 = new EgeDeniziB("Karagöz Balığı", "Karagöz balığının Çipura ile yakın akrabalığı bulunmaktadır. Sarmos, mırmır, sivri gaga gibi birkaç çeşidi vardır.Maksimum 50 cm boyutuna ulaşabilir. Ülkemizde Akdeniz, Karadeniz ve Ege Denizi bölgelerimizde yaşarlar. Bu bölgelerde yaygın olarak 25 cm civarında görülür. Kayalık, kumlu alanlarda yaşarlar.Görünüşü ile Çupra balığına benzemesinin yanı sıra lezzet olarak da benzerdir. Serin dönemlerde tüketimesi tavsiye edilir bu dönemlerde eti daha yağlı ve lezzetli olmaktadır.");  EgeDeniziB balık16 = new EgeDeniziB("Kolyoz Balığı", "Uskumsugillerden olan kolyoz, Scombridae familyasından bir deniz balığıdır. Uskumruya çok benzeyen bir balıktır. Kuyruk yüzgecine bakılarak ayırt edilebilir. Bu türün kuyruk yüzgecinin ucu daha sivridir.Genç olanlar kıyılara yakın kumlu alanlar ve yosun yataklarında yaşarken, yetişkin olanlar daha açıkta derin sularda yaşar.Sürüler halinde yaşamlarını sürdürürler. En fazla 50 cm uzunluğa kadar büyüyebilir ancak ülkemiz sularında görülen yetişkinler ortalama 20-25 cm civarındadır. Eti lezzetli ege balıkları arasındadır.");  EgeDeniziB balık17 = new EgeDeniziB("Kırma Mercan", "Kırma mercan balığı (Pagellus erythrinus) çipura ailesinden, Sparidae familyası mensubu, Akdeniz ve Ege balıklarından biridir. Özellikle Akdeniz ülkelerinde bolca tüketilen lezzetli bir balık türüdür.İnce, oval yapıda bir vücuda sahip olan kırma mercan en fazla 50 cm boyutuna ulaşabilir. Genel olarak 15-30 cm arasında görülürler. Bu tür bir hermafrodittir; sonradan cinsiyet değiştirebilirler.Genellikle hayatlarının ilk yıllarında dişi, sonraki yıllarında ise erkek olurlar. Hepçil bir balıktır ve ana besinleri küçük balıklar ve omurgasızlardan oluşur. Akdeniz ülkelerinde ticari değere sahip, lezzetli bir balıktır.");  EgeDeniziB balık18 = new EgeDeniziB("Lahoz Balığı", "Lahoz balığı Hani balıkları ailesinden, Ege ve Akdeniz bölgelerinde bulunan bir türdür. Bu tür iri ege balıkları arasındadır. En fazla 125 cm uzunluk ve 25 kg ağırlığa ulaşabilirler.Yaşam alanları 20-250 metre arasında kayalık, çakıllı ve taşlı alanlardır. Son derece yırtıcı etçil balıklardan biridir. Yiyebileceği her türlü balık, omurgasız ve kabuklu türleri ile beslenir.Orfoz balığı ile yakından akrabadır ve ticari değeri bulunduğu bölgeye göre değişir. Özellikle Akdeniz bölgesinde etinin lezzetli ile bilinir.Bu tür birkaç farklı isimle bilinir.");  EgeDeniziB balık19 = new EgeDeniziB("Levrek", "Birçoğumuzun da yakından tanıdığı Levrek balığı, Dicentrarchus familyasından bir balık türüdür. Bilimsel olarak ilk kez 1758 yılında Dicentrarchus labrax olarak tanımlanmıştır.Şimdiye kadar kayıt altına alınan en büyük boyutları 1m ve 12 kg olsa da yaygın olarak 50 cm ve 5 kg civarında görülürler. Akdeniz ve Ege başta olmak üzere tüm denizlerimizde yaşadıkları biliniyor.Haliçler, lagünler, akarsuların denize döküldüğü yerler ana yaşam alanlarıdır. Kısa bir süreliğine tatlı sulara geçtikleri de bilinmektedir (tatlı su levreklerinin dışında).Ticari değeri yüksek, etinin lezzetini kanıtlamış Ege denizi balık türlerinden biridir.");  EgeDeniziB balık20=new EgeDeniziB("Lipsoz Balığı", "Lipsoz balığı, görünümü ile iskorpit balığına benzeyen Scorpaenidae familyasından bir balık türüdür. Lipsos ismiyle de bilinen bu türün bilimsel adı Scorpaena scrofa’dır.Ülkemiz sularında Akdeniz ve Ege Denizi türlerinden biridir, Karadeniz bölgesinde bulunmaz. S. porcus türü Karadeniz bölgesinde görülebilir. İskorpit balığı gibi bu balık da zehirli balıklardan biridir.Vücut rengi kiremit rengi tonlarından, pembemsi tonlara kadar değişebilir. En fazla 50 cm ve 3 kg ağırlığa ulaştığı biliniyor ancak, sularımızdaki yetişkin lipsoz balıkları genellikle 25-30 cm civarındadır.Yenilebilir balıklardan biridir, en çok çorba ve buğulama yapıldığında lezzetlidir.");  EgeDeniziB balık21 = new EgeDeniziB("Lüfer Balığı", "Lüfer, Pomatomidae familyasından bir balık türüdür. Ege balıkları arasında ekonomik değeri yüksek ve çok lezzetli bir balıktır. Bilimsel adı Pomatomus saltatrix olan lüfer, ülkemizin tüm denizlerinde bulunmaktadır.Maksimum ölçülen boyutları 120 cm ve 14 kg’dır. Bölgemizde yaşayan yetişkin lüferler 30-60 cm arasında bulunurlar. Lüfer, gevşek ve küçük gruplar halinde yaşayan yırtıcı deniz balıklarından balıklardan biridir.Lüfer çeşitli boyutlara göre farklı isimler almıştır:Sarıkanat: 18 – 25 cm boyutlarında olan lüfer yavrusudur. 18 cm altının avlanması veya satılması yasaktır.Lüfer: 28 – 35 cm boyutlarına ulaştığında lüfer adı verilir. Avlamak ve satış serbesttir.Kofana: 35 cm üzerine çıkan lüferlere bu ad verilmiştir. Avlanması ve satılması serbesttir.Sırtıkara: 50 cm’nin üzerindeki lüferlere verilmiş isimdir. Ülkemiz denizlerinde uzun süredir görülmemiştir. Avlamak ve satışını yapmak serbesttir.Lüfer bir dönem koruma altına alınarak avlanılması ve satışı yapılması yasaklanmış balık türlerindendir. Ancak, günümüzde yasak bitirilerek 18 cm ve üzeri olanların avlanması ve satılması serbest bırakılmıştır.");  EgeDeniziB balık22 = new EgeDeniziB("Mahmuzlu camgöz köpek balığı", "Mahmuzlu camgöz Squalidae familyasından, Akdeniz ve Ege Denizi başta olmak üzere ülkemiz sularında bulunmaktadır. Ülkemizde tüketilmese de Avrupa ülkelerinde tüketildiği bilinmektedir.Ana besinleri balık sürüleri ve ahtapotlar olan bu türün, profesyonel balıkçıların ağlarına ciddi hasar verdiği biliniyor. Nadiren de olsa dip balıklarını avlayan amatör balıkçıların oltasına takıldığı görülmektedir.En fazla 150 cm ve 10 kg boyutlarına eriştikleri tespit edilmiştir ancak, genellikle 80 cm ve 4 kg civarlarına ulaşırlar. Ticari değeri yoktur.");  EgeDeniziB balık23 = new EgeDeniziB("Mandagöz Mercan Balığı", "Kırmızı (kızıl) çipura olarak da bilinen mandagöz mercan (Pagellus bogaraveo) Sparidae familyasından bir deniz balığıdır.Akdeniz başta olmak üzere ılıman ve sıcak denizleri tercih eden bir türdür, ılıman ısısıyla Ege balıklarından da biridir. Bulunduğu bölgeye göre en fazla 400 ila 700 metre derine inebilen bir balıktır.Kaydedilen en büyük uzunluk 70 cm, yaygın olarak da 30 cm ve 4 kg’dır. Eti lezzetli, ızgarada pişilmesi tercih edilen bir mercan varyantıdır.");  EgeDeniziB balık24 = new EgeDeniziB("Melanur Balığı", "Melanur diğer adıyla Melanurya, Sparidae familyasından bir deniz türüdür. Ülkemizde Akdeniz, Marmara ve Ege bölgerinde bulunur. Bilimsel adı Oblada melanura olan melanur, Ege denizi balık türleri arasında yer alıyor.Çok iri boyutlara ulaşabilen bir tür değildir. Kaydedilen maksimum boyutları, 38 cm ve 930 gr’dır. Ülkemiz sularında yaygın olarak 20 cm civarında görülürler. Hepçil olan türün ana besini omurgasızlardır.Eti lezzetli türlerden biri olan melanurun ticari değeri orta seviyelerdedir.");  EgeDeniziB balık25 = new EgeDeniziB("Mersin Balığı", "Mersin balığı birçok farklı türün ortak adıdır. Mersin balığı adı altında 19 farklı cins balık bulunmaktadır. Görünümlerindeki ufak farklılar ile ayrılırlar. Bu balıklar Acipenseridae familyasına aittir. Akdeniz bölgemizde daha yaygın olan tür, Ege balıkları arasında da bulunmaktadır.Türe göre boyutları farketmektedir. Yetişkin mersin balıkları ortalama 140 ila 300 cm uzunluğa, 100 ila 200 kg ağırlığa ulaşabilirler. Şimdiye kadarki ölçülen en büyük boyutlar; 7.2 metre ve 1571 kilogramdır.Mersin balığı altında bulunan 19 farklı tür şu şekildedir:Sibirya mersini ,Kısa burunlu mersin balığı,Yangtze mersin balığı,Göl mersin balığı,Rus mersin balığı,Yeşil mersin balığı,Sakhalin mersin balığı,Japon mersin balığı,Adriyatik mersin balığı,Şip balığı,Körfez mersinbalığı,Atlantik mersin balığı,İran mersin balığı,Çuka balığı,Amur mersin balığı,Çin mersin balığı,Yıldızlı mersin balığı,Kolan balığı,Beyaz mersin balığı");  EgeDeniziB balık26=new EgeDeniziB("Mürekkepbalığı", "Mürekkepbalığı Cephalopoda (Kafadanbacaklılar) sınıfından, deniz türleri arasında olan bir yumuşakça türüdür. İkisi diğerlerinden daha uzun olan toplam 10 adet kolları vardır ve iç bölgelerinde çok sayıda vantuz bulunur.Ege denizi balıkları arasında yer alan mürekkepbalığı, sıcak sularda yaşayan bir türdür. Bu türün boyutları çok değişkendir. Cinse göre 20 cm ile 17 metre arasında değişen çeşitleri vardır. Yaygın olarak yetişkin olanları 50-60 cm arasındadır.Ticari değeri olan, eti lezzetli bir yumuşakça türüdür.");  EgeDeniziB balık27 = new EgeDeniziB("Mırmır Balığı", "Sparidae familyasından olan mırmır balığı, ekonomik değeri yüksek lezzetli balık türlerinden biridir. Denizin diplerinde ve kumlu kısımlarda yaşayan mırmır balığı sıklıkla avlanan Ege balıkları arasındadır.Sparidae familyasından olan mırmır balığı, ekonomik değeri yüksek lezzetli balık türlerinden biridir. Denizin diplerinde ve kumlu kısımlarda yaşayan mırmır balığı sıklıkla avlanan Ege balıkları arasındadır.En fazla 55 cm ve 1 kg ağırlığa ulaşmaktadırlar ancak, yaygın boyutları 30 cm civarındadır. Ticari değeri yüksek Ege denizi balık türlerinden biridir.");  EgeDeniziB balık28=new EgeDeniziB("Orfoz Balığı", "Serranidae familyasından Orfoz, Taş hanisi adıyla da bilinir. Bilimsel adı Epinephelus marginatus’dur. Orfozlar hermafrodit (çift cinsiyetli) deniz balıkları arasındadır.Orfoz balığı neslinin tükenmesiyle karşı karşıya olduğu için avlanması yasak türlerdendir. Ülkemizde Akdeniz ve Ege denizlerinin güney kısımlarında yaşarlar.En fazla 140 cm ve 60 kg boyutlarına ulaşabilirler. Boyutları ile iri ege balıklarından biridir. Ülkemiz denizlerinde yaygın görülen boyutları ortalama 60 cm ve 15 kg’dır.");  EgeDeniziB balık29 = new EgeDeniziB("Orkinos", "Çoğumuzun marketlerde konserve içerinde gördüğü Orkinos, Uskumrugiller (Scombridae) ailesinin üyesidir. Kendi aralarında farklı türlere sahip olan, Orkinos ortak adı verilen bu balık denizlerdeki en iri balıklardan biridir.Ege denizi balıkları arasında en büyük türlerden biridir. Yetişkin bir ton balığı, 6 metre uzunluğa ve 1 ton ağırlığa ulaşmaktadır. Ancak, bu boyutlarda Orkinos bulma ihtimali çok düşüktür genellikle 3-4 metre ve 400-600 kg arasında avlanırlar.Ekonomik değeri çok yüksek ve son deree lezzetli balıklardandır.");  EgeDeniziB balık30 = new EgeDeniziB("Pisi Balığı", "Pisi balığı kalkan ile karıştırılabilir ancak resimlerine bakıldığında kolayca ayırt edilebilir. Pleuronectidae familyasından olan pisi balığının vücudu kalkana göre daha elips şeklindedir ve kalkanın sırtında olan düğme diye tabir edilen kemikli yapılar yoktur.Yine pisi balığının gözleri de vücudun sağ tarafındadır. En fazla 60 cm boyut ölçülmüştür ancak, yaygın boyutları 30 cm civarındadır.Ülkemizde Akdeniz, Ege, Karadeniz ve Marmara bölgelerimizde bulunur.");  EgeDeniziB balık31 = new EgeDeniziB("Sardalya Balığı", "Saldalya hamsi ile yakından akraba, Clupeinae familyasından bir balık türüdür. Sürüler halinde kıyıya yakın geçerek göç ederek yaşamlarını sürdürürler.Akdeniz ve Karadeniz’de daha yaygın görülen sardalya, Ege denizi balık türleri arasında da yer almaktadır. Denizlerimizde 15-20 cm boyutlara ulaşırlar ancak okyanus bölgelerindeki sardalyalar 30 cm uzunluğa kadar büyümektedir.Ticari değeri çok yüksek, lezzetli balık türlerindendir.");  EgeDeniziB balık32 = new EgeDeniziB("Sargan Balığı", "Tipik balık görünümünün dışında uzun ve ince bir vücuda sahip sargan (Belone belone), Belonidae familyasının üyesidir. Vücut yapıları sayesinde hızlı ve çevik deniz balıklarındandır.Yaşadıkları bölgeye göre 1 metre uzunluğa erişmektedirler ancak Akdeniz ve Ege denizlerimizde 60 cm civarına kadar büyümektedirler.Ana besinleri küçük balıklardır ve hamsi, çaça gibi balık türleri ile beslenir. Ticari değeri yüksek, lezzetli Akdeniz ve Ege balıkları arasındadır.");  EgeDeniziB balık33=new EgeDeniziB("Sargoz Balığı", "Sargoz, Akdeniz ve Ege bölgelerinde sıklıkla Karagöz ile karıştırılır. Diplodus sargus bilimsel adı verilmiştir ve Sparidae ailesinden bir deniz balığı türüdür.Bu balık bölgeye göre farklı isimler almıştır. Baltabaş, Sargos ve Ak Karagöz olarak da bilinirler. Ülkemizde Akdeniz ve Ege bölgelerimizde yaygındır.Güçlü çeneye sahiptirler ana besinleri; kabuklular, yumuşakçalar ve deniz yosunları. Ekonomik değeri yüksek, lezzetli bir balıktır.");  EgeDeniziB balık34 = new EgeDeniziB("Sinarit Balığı", "Sparidae familyasından olan Sinarit’in bilimsel adı Dentex dentex’dir. Akdeniz’de yaygındır ancak Karadeniz, Marmara ve Ege balıkları arasında da yer alırlar.Taşlı ve kumlu bölgelerde 200 metre derinliğe kadar yaşarlar. Ana besinleri kafadanbacaklılar ve yumuşakçalardır. Genellikle yalnız yaşarlar ancak üreme dönemlerinde küçük sürüler halinde görülmektedirler.1 metre uzunluğa, 20 kg ağırlığa ulaşırlar. Olta avcılığı son derece zevkli, eti lezzetli ama az bulunan bir balıktır.");  EgeDeniziB balık35 = new EgeDeniziB("Tekir Balığı", "Bilimsel adı Mullus surmuletus olan Tekir, Mullidae familyasından bir deniz balığıdır. Akdeniz, Karadeniz ve Ege denizi balık türleri arasındadır.5 metrelik sığ sulardan 400 metre derinlikteki sulara kadar uzanan yaşam alanları vardır. Kaydedilmiş en büyük boyutları 40 cm ve 1 kg’dır ancak sularımızda yaygın olarak 25 cm civarında görülürler. Ticari değeri olan bir balıktır ve av balığı olarak da kullanılmaktadır.");  EgeDeniziB balık36 = new EgeDeniziB("Trakonya Balığı", "Trakonya balığı halk arasında dragon balığı olarak da bilinmektedir. Trachinidae ailesinden olan balığın bilimsel adı Trachinus draco’dur.Vücudunda zehirli dikenlere sahiptir ve dokunulduğunda toksik bir madde salgılar. Zehirli Ege balıkları arasındadır. Zehri kuvvetlidir, uzuv kaybı veya kalp krizine neden olabilmektedir.1 metre ile 150 metre arasında kumlu, çakıllı alanlarda yaşamaktadır. Küçük balıklar, omurgasızlar ve kabuklular ana besinleridir. En fazla 55 cm boyuta ulaşabilirler ve genellikle 25 cm civarında görülürler. Ticari değer yoktur.");  EgeDeniziB balık37 = new EgeDeniziB("Trança", "Ege balıkları arasında meşhurlaşmış türlerden biridir. Bölgede Çavuş, Antenli mercan, Altınkaş isimleri ile de bilinir. Bilimsel adı Pagrus caeruleostictus’dur ve mercan familyasından bir türdür.Sığ ve sıcak sularda tek olarak yaşayan iri bir balıktır. Etçil bir balıktır ve kafadanbacaklılar, küçük balıklar ana besinleridir. Okyanuslara ve farklı denizlere sürekli göç eden balıklardandır.Yetişkin boyutları 60-75 cm civarındadır ancak 1 metre üzerine de çıkabilmektedir. Ekonomik değeri yüksek, lezzetli bir balıktır.");  EgeDeniziB balık38=new EgeDeniziB("Zurna Balığı", "Zurna balığı (Scomberesox saurus), Scomberesocidae familyasından bir deniz balığıdır. Açık denizlerde yaşayan bu tür, Süveyş kanalının açılmasından sonra Akdeniz ve Ege denizlerinde görülmüştür.En fazla 50 cm, ortamala 35 cm boyutlarındadır. Balık yavruları ve plankton ile beslenir. Avlanmadıkları için herhangi bir ticari değeri yoktur.");  class Node1  {  public EgeDeniziB data;  public Node1 left;  public Node1 right;  public Node1(EgeDeniziB data)  {  this.data = data;  left = null;  right = null;  }  }  internal class BalıkAgacı  {  public Node1 root;  public Node1 getRoot()  {  return root;  }  public BalıkAgacı()  {  root = null;  }  public Node1 newNode(EgeDeniziB data)  {  root = new Node1(data);  return root;  }  public Node1 insert(Node1 root, EgeDeniziB data)  {  Node1 eleman = new Node1(data);  if (root != null)  {  if (data.balıkAdı.ToLower().CompareTo(root.data.balıkAdı.ToLower()) < 0)  {  root.left = insert(root.left, data);  }  else  {  root.right = insert(root.right, data);  }  }  else  {  root = newNode(data);  }  return root;  }  public void BuildBalancedTree(EgeDeniziB[] sortedBalıklar)//1.d için  {  this.root = BuildBalancedTreeRecursive(sortedBalıklar,0,sortedBalıklar.Length-1);  }  public Node1 BuildBalancedTreeRecursive(EgeDeniziB[] sortedBalıklar, int startIndex, int endIndex)//1.d içinıı  {  if (startIndex>endIndex)  {  return null;  }  int mid=(startIndex+endIndex)/2;  Node1 node = newNode(sortedBalıklar[mid]);  node.left = BuildBalancedTreeRecursive(sortedBalıklar, startIndex, mid - 1);  node.right=BuildBalancedTreeRecursive(sortedBalıklar, mid + 1, endIndex );  return node;  }    //orderlarda balık adını yazdırıyoruz.  public void preOrder1(Node1 root)  {  if (root != null)  {  Console.WriteLine(root.data.balıkAdı + " ");  preOrder1(root.left);  preOrder1(root.right);  }  }  public void inOrder1(Node1 rooot)//1.a kısmı burada komple halledildi  {  if (rooot != null)  {  inOrder1(rooot.left);  Console.WriteLine("mevcut balık: "+rooot.data.balıkAdı);  Console.WriteLine(rooot.data.balıkAdı+" balığının kelime ağacının inorder sıralanması: ");  rooot.data.kelimelerAğacı.inOrder(rooot.data.kelimelerAğacı.root);  Console.WriteLine("\n");  inOrder1(rooot.right);  }    }  public void postOrder1(Node1 root)  {  if (root != null)  {  postOrder1(root.left);  postOrder1(root.right);  Console.WriteLine(root.data.balıkAdı + " ");  }  }  public int size(Node1 root)  {  if (root == null)  {  return 0;  }  else  {  return size(root.left) + 1 + size(root.right);  }  }  public int height(Node1 root)  {  if (root == null)  {  return -1;//yükseklik yok!  }  else  {  int l, r;  l = height(root.left);  r = height(root.right);  return Math.Max(l, r) + 1;  }  }  }  EgeDeniziB[] balıkList = new EgeDeniziB[] {balık1,balık2,balık3,balık4,balık5,balık6,balık7,balık8,balık9,balık10,balık11,balık12,balık13,balık14,balık15,balık16,balık17  ,balık18,balık19,balık20,balık21,balık22,balık23,balık24,balık25,balık26,balık27,balık28,balık29,balık30,balık31,balık32,balık33,balık34,balık35,balık36,balık37,balık38  };  BalıkAgacı balıkAgacı=new BalıkAgacı();  foreach (EgeDeniziB item in balıkList)  {  balıkAgacı.root = balıkAgacı.insert(balıkAgacı.root,item);  }//balıkAğacı initialize işlemi tamamlandı.  Console.WriteLine("1.a kısmı "); //1.a kısmı  balıkAgacı.inOrder1(balıkAgacı.root);  Console.WriteLine();  Console.WriteLine("--------------------------------------------------------------");  Console.WriteLine("1.b kısmına geçmek için enter'a basın");  Console.ReadLine();  public void inOrder1(Node1 rooot)//1.a kısmı burada komple halledildi  {  if (rooot != null)  {  inOrder1(rooot.left);  Console.WriteLine("mevcut balık: "+rooot.data.balıkAdı);  Console.WriteLine(rooot.data.balıkAdı+" balığının bilgi paragrafının alfabetik sıralanması: ");  rooot.data.kelimelerAğacı.inOrder(rooot.data.kelimelerAğacı.root);  Console.WriteLine("\n");  inOrder1(rooot.right);  }    } |

### 1.a.2 Açıklama

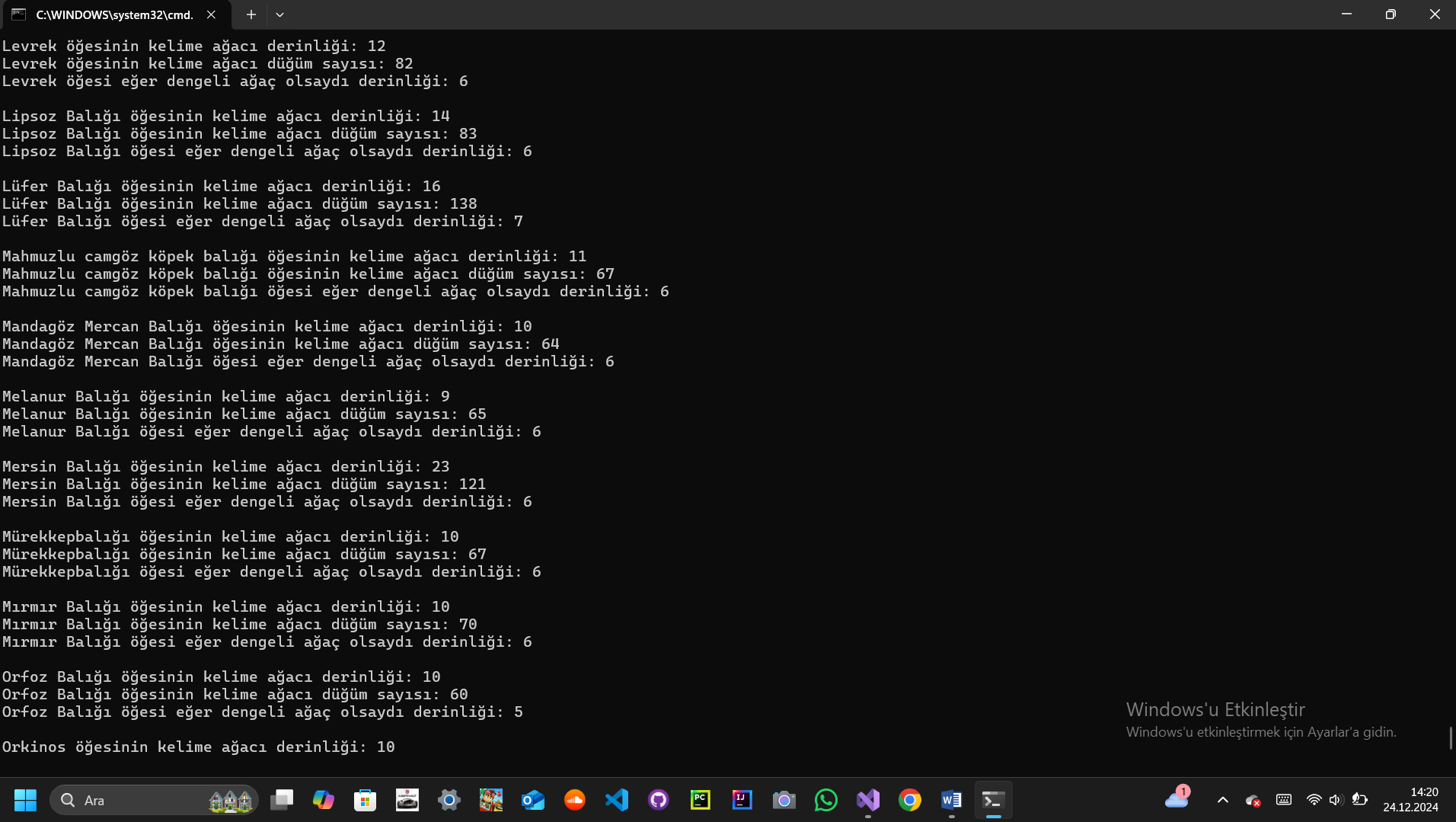
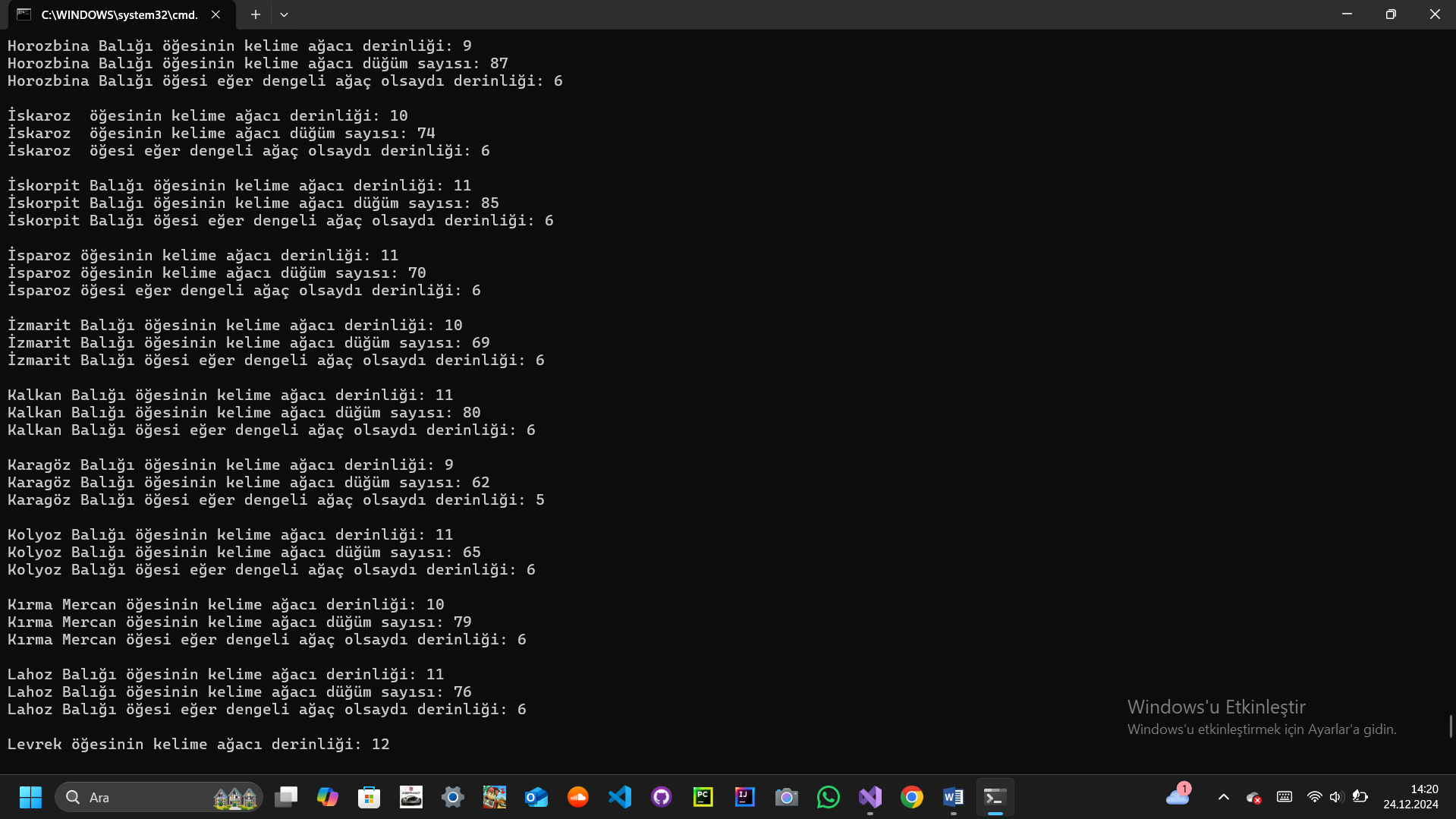
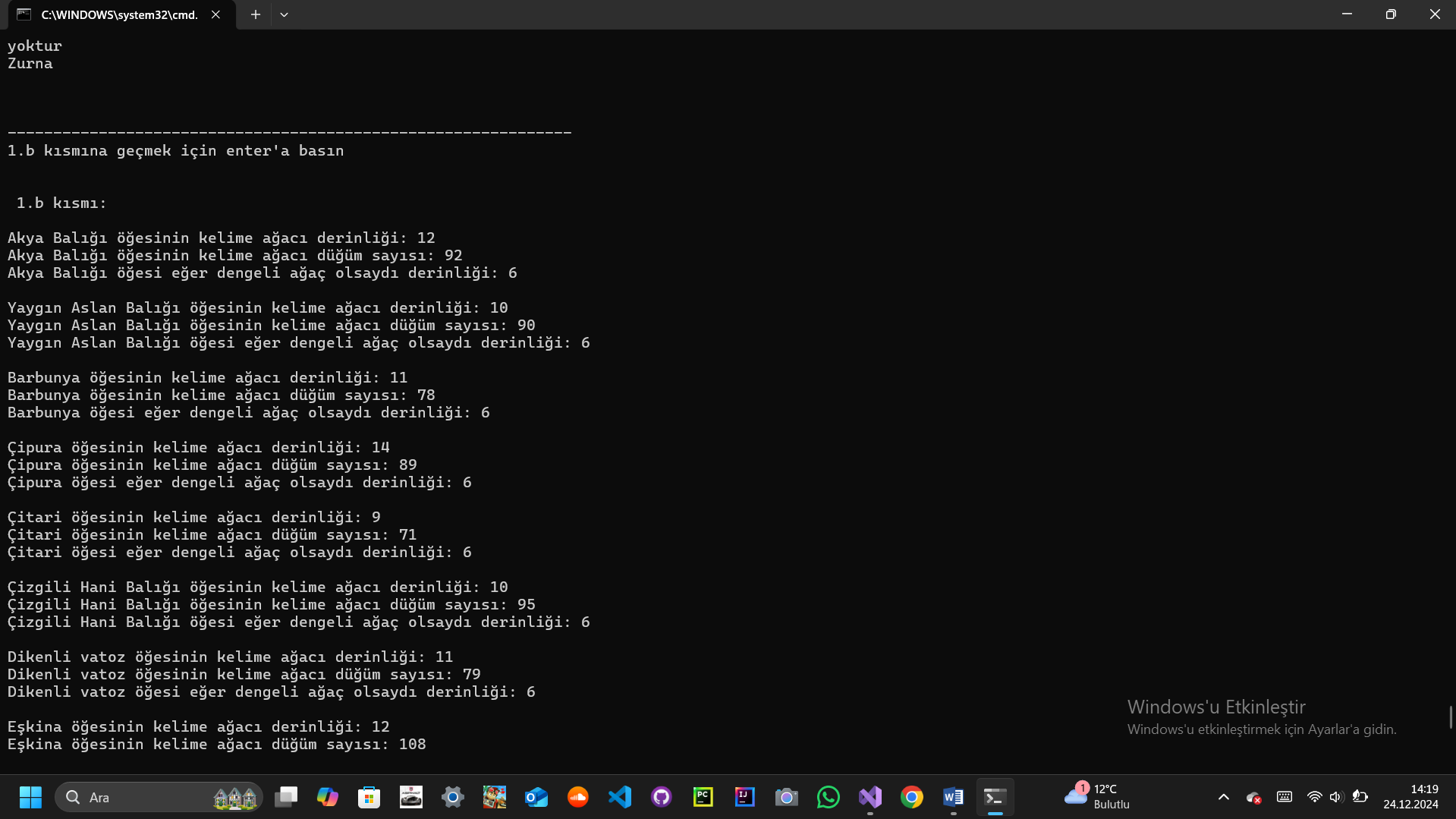
İlk başta istenildiği gibi kelimelerağacı, EgeDeniziB sınıfı ve BalıkAğacı sınıfları oluşturuldu(kodda da bunlara bakabilirsiniz).Kelimeler ağacına 50, 100 gibi rakamları, noktalama ve ayırma işaretlerini (genel olarak kelime harici hiçbir şeyi) eklemedim. Ayrıca kelimeleri karşılaştırırken küçük harfe çevirip karşılaştırdım(ToLower() metoduyla). Daha sonra internet sitesinden istenildiği üzere balık nesneleri oluşturulup ağaca eklendi ve inorder1() metoduyla balık adları, içindeki kelimeler ağacıyla beraber alfabetik olarak listelendi.

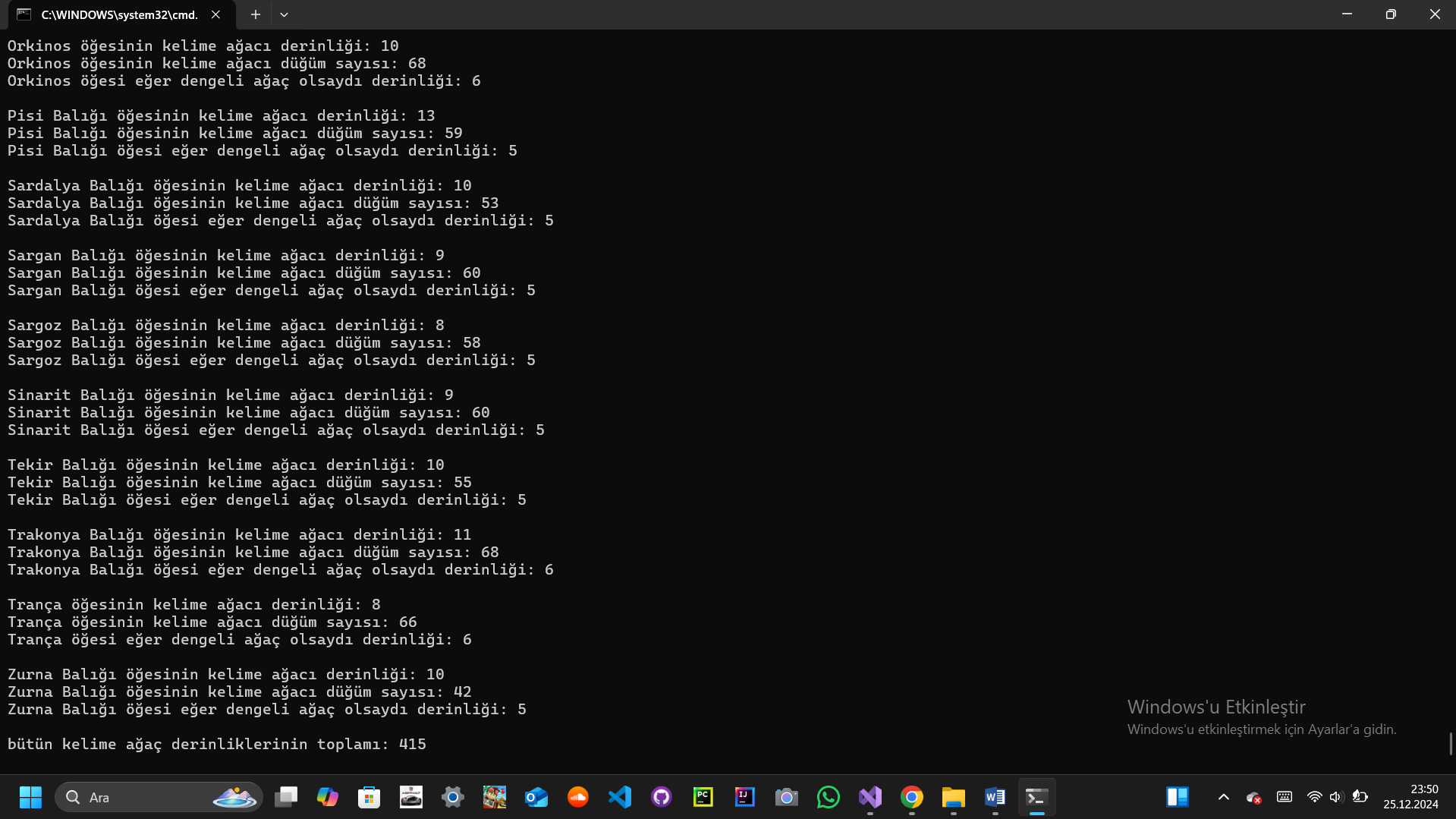
## 1.b Derinlik Bulma, Ağacı Listeleme, Düğüm Sayısı Buldurma, Dengeli Ağaç Derinliği Hesaplama

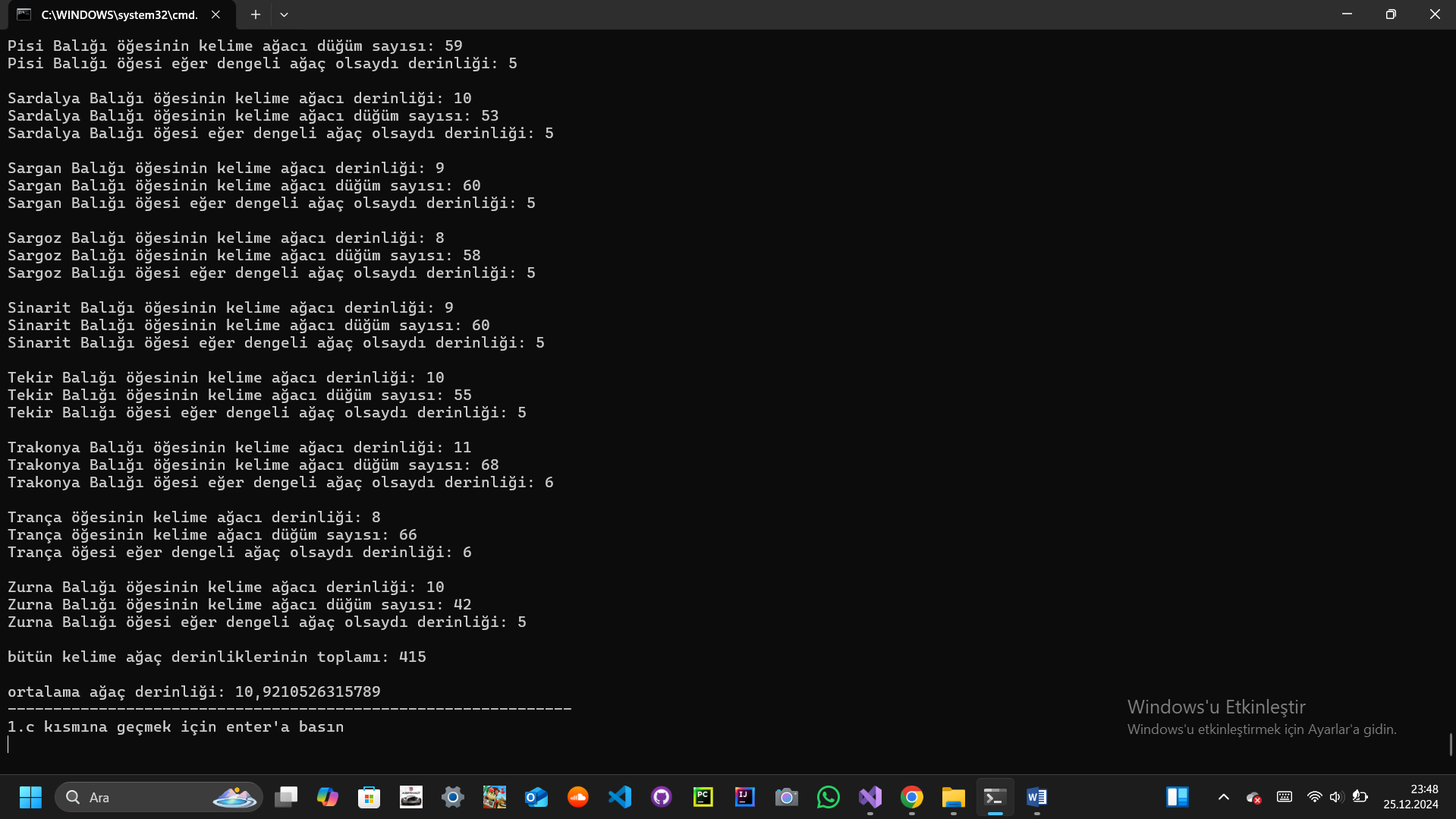
### 1.b.1 Kodlar

|  |
| --- |
| Console.WriteLine("\n 1.b kısmı: " );//1.b kısmı  double sayaç2 = 0;  int balancedDepth=0 ;  for (int i = 0; i < balıkList.Length; i++)  {  for (int j = 0; j < Math.Pow(2,9) ; j++)//düğüm sayısının 2 nin hangi kuvvetleri arasında olduğuna buluyoruz.  {  if (balıkList[i].kelimelerAğacı.size(balıkList[i].kelimelerAğacı.root)>=Math.Pow(2,j) && (balıkList[i].kelimelerAğacı.size(balıkList[i].kelimelerAğacı.root) < Math.Pow(2, j+1)) )  {  balancedDepth = j;  }  }  Console.WriteLine();  Console.WriteLine(balıkList[i].balıkAdı+" öğesinin kelime ağacı derinliği: " + balıkList[i].kelimelerAğacı.height(balıkList[i].kelimelerAğacı.root));  Console.WriteLine(balıkList[i].balıkAdı + " öğesinin kelime ağacı düğüm sayısı: " + balıkList[i].kelimelerAğacı.size(balıkList[i].kelimelerAğacı.root));  Console.WriteLine(balıkList[i].balıkAdı + " öğesi eğer dengeli ağaç olsaydı derinliği: " +balancedDepth);  sayaç2 = sayaç2 + balıkList[i].kelimelerAğacı.height(balıkList[i].kelimelerAğacı.root);  }  Console.WriteLine("\nbütün kelime ağaç derinliklerinin toplamı: "+sayaç2);  Console.WriteLine("\nortalama ağaç derinliği: "+sayaç2/balıkList.Length);  Console.WriteLine("--------------------------------------------------------------");  Console.WriteLine("1.c kısmına geçmek için enter'a basın");  Console.ReadLine();  public int size(Node root)  {  if (root == null)  {  return 0;  }  else  {    return size(root.left)+1+size(root.right);  }  }  public int height(Node root)  {  if(root == null)  {  return -1;//yükseklik yok!  }  else  {  int l, r;  l=height(root.left);  r=height(root.right);  return Math.Max(l, r)+1;  }  } |

### 1.b.2 Ekran görüntüleri







### 1.b.3 Açıklama

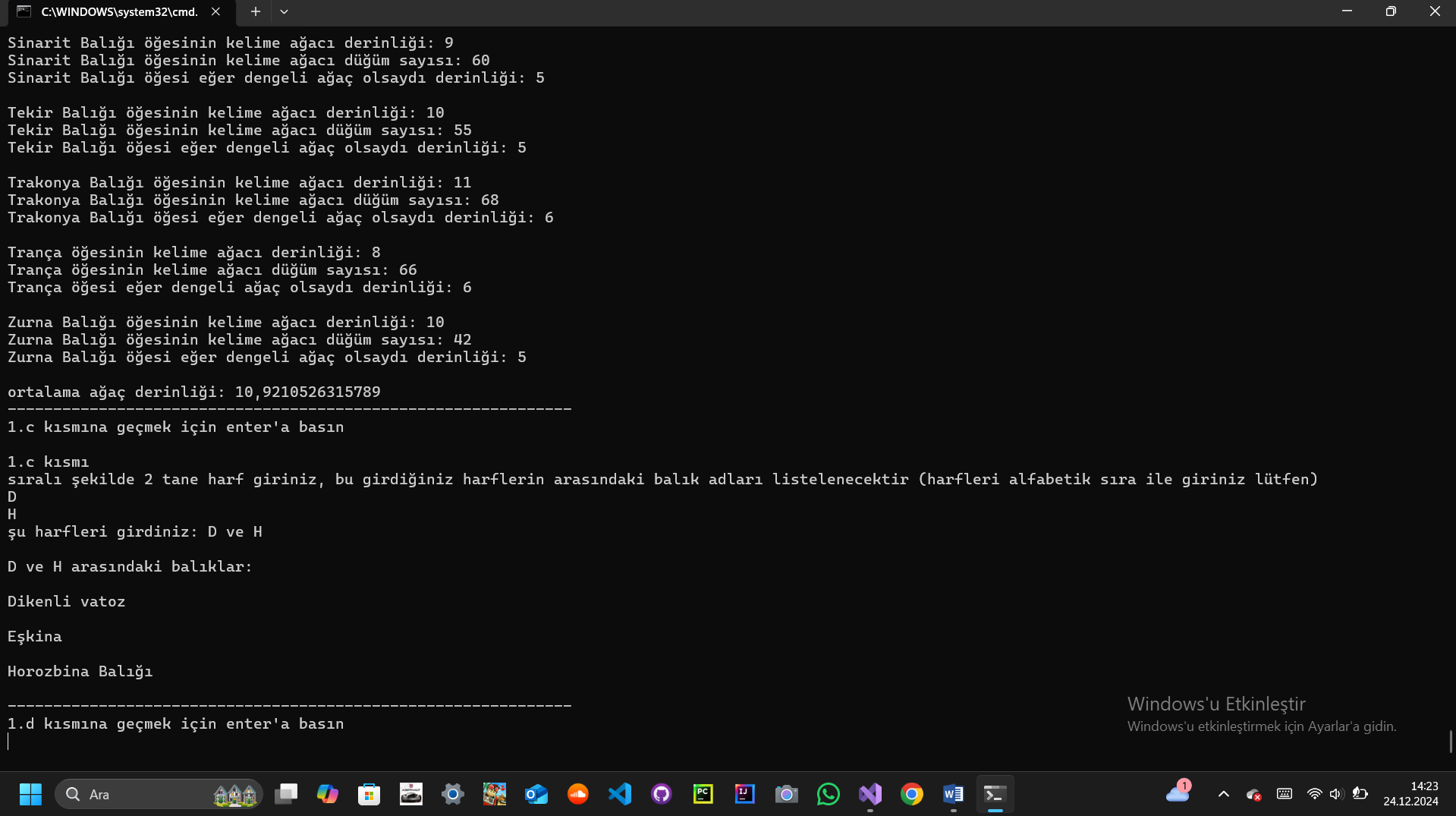
Oluşturduğumuz balıklar ağacındaki EgeDeniziB nesnelerini tek tek dolaşıp bunların derinlik ve düğüm sayılarını buluyoruz, düğüm sayısının 2’nin hangi kuvvetleri arasında olduğuna bakarak dengeli ağaç olsalardı derinliklerinin ne olacağını buluyoruz.( Daha detaylı inceleme için koda göz atabilirsiniz. )

## 1.c Arama Listeleme

### 1.c.1 Kodlar

|  |
| --- |
| Console.WriteLine("1.c kısmı");//1.c kısmı  Console.WriteLine("sıralı şekilde 2 tane harf giriniz, bu girdiğiniz harflerin arasındaki balık adları listelenecektir (harfleri alfabetik sıra ile giriniz lütfen)");  string h1=Console.ReadLine();  string h2=Console.ReadLine();  Console.WriteLine("şu harfleri girdiniz: "+h1+" ve "+h2+"\n");  Console.WriteLine(h1+" ve "+h2+ " arasındaki balıklar:\n ");  balıklarAğacı.istenenAralık(balıklarAğacı.root,h1,h2);  Console.WriteLine("--------------------------------------------------------------");  Console.WriteLine("1.d kısmına geçmek için enter'a basın");  Console.ReadLine();  public void istenenAralık(Node root,string h1,string h2)//1.c kısmı için fonksiyon:hocanın istediği şekilde yapmak için büyük olan stringin(h2) ilk harfini alıp kıyasladım.  {  if (root != null)  {  istenenAralık (root.left,h1,h2);    if (root.data.ToLower().CompareTo(h1.ToLower()) >=0 && root.data.Substring(0,1).ToLower().CompareTo(h2.ToLower())<=0)  {  Console.WriteLine(root.data+"\n");  }  istenenAralık(root.right,h1,h2);  }  } |

### 1.c.2 Ekran görüntüleri

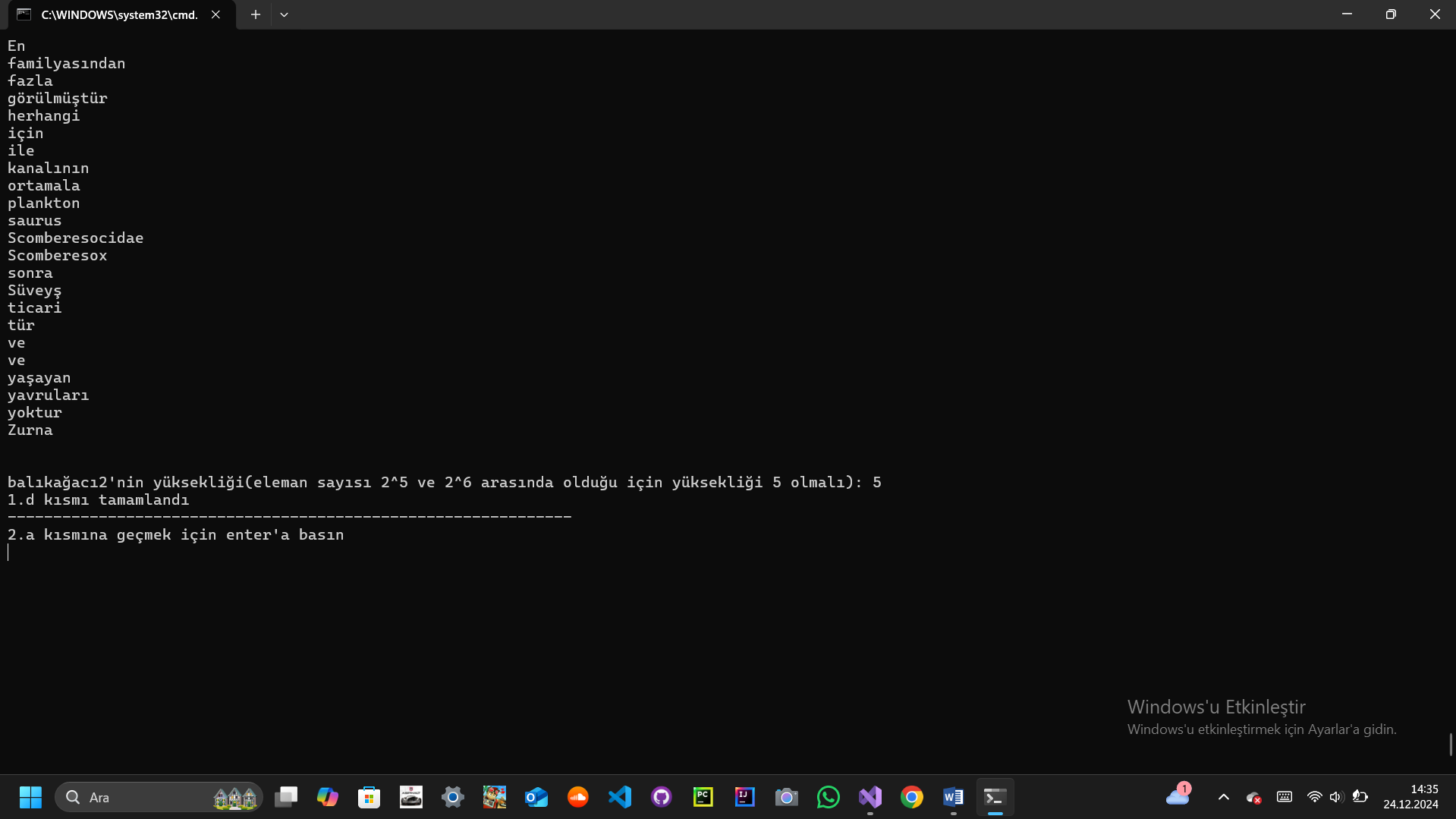
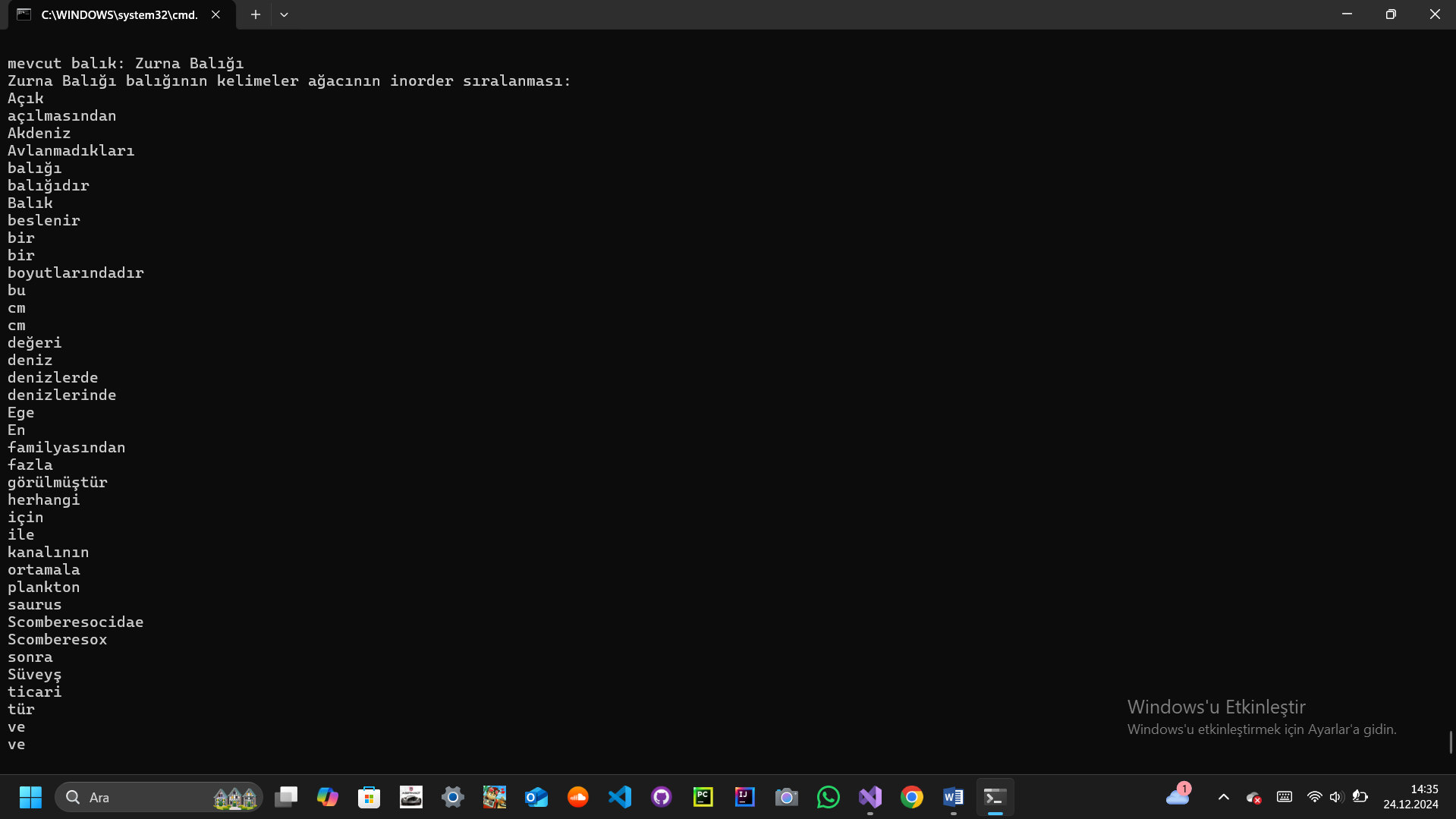
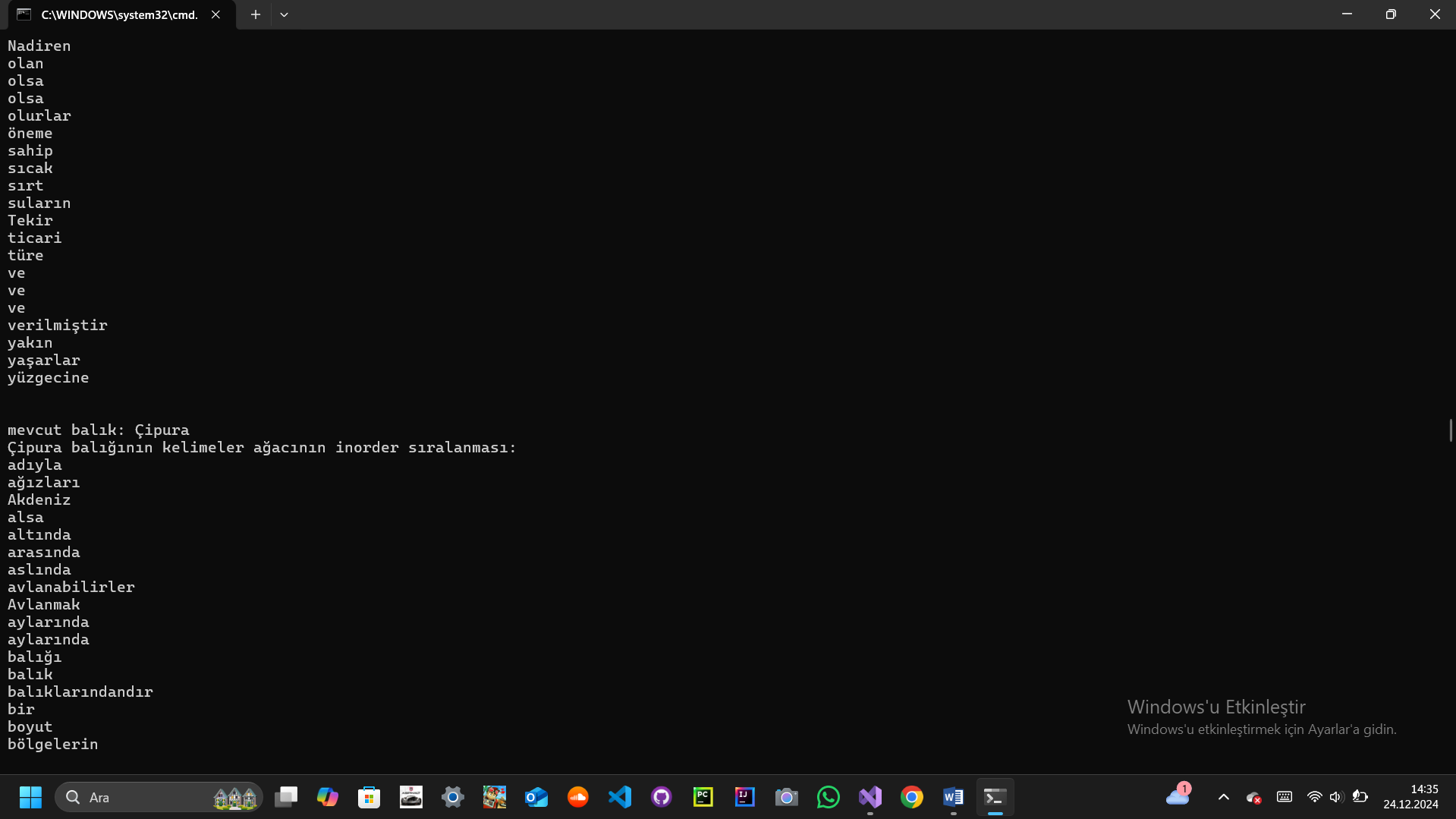
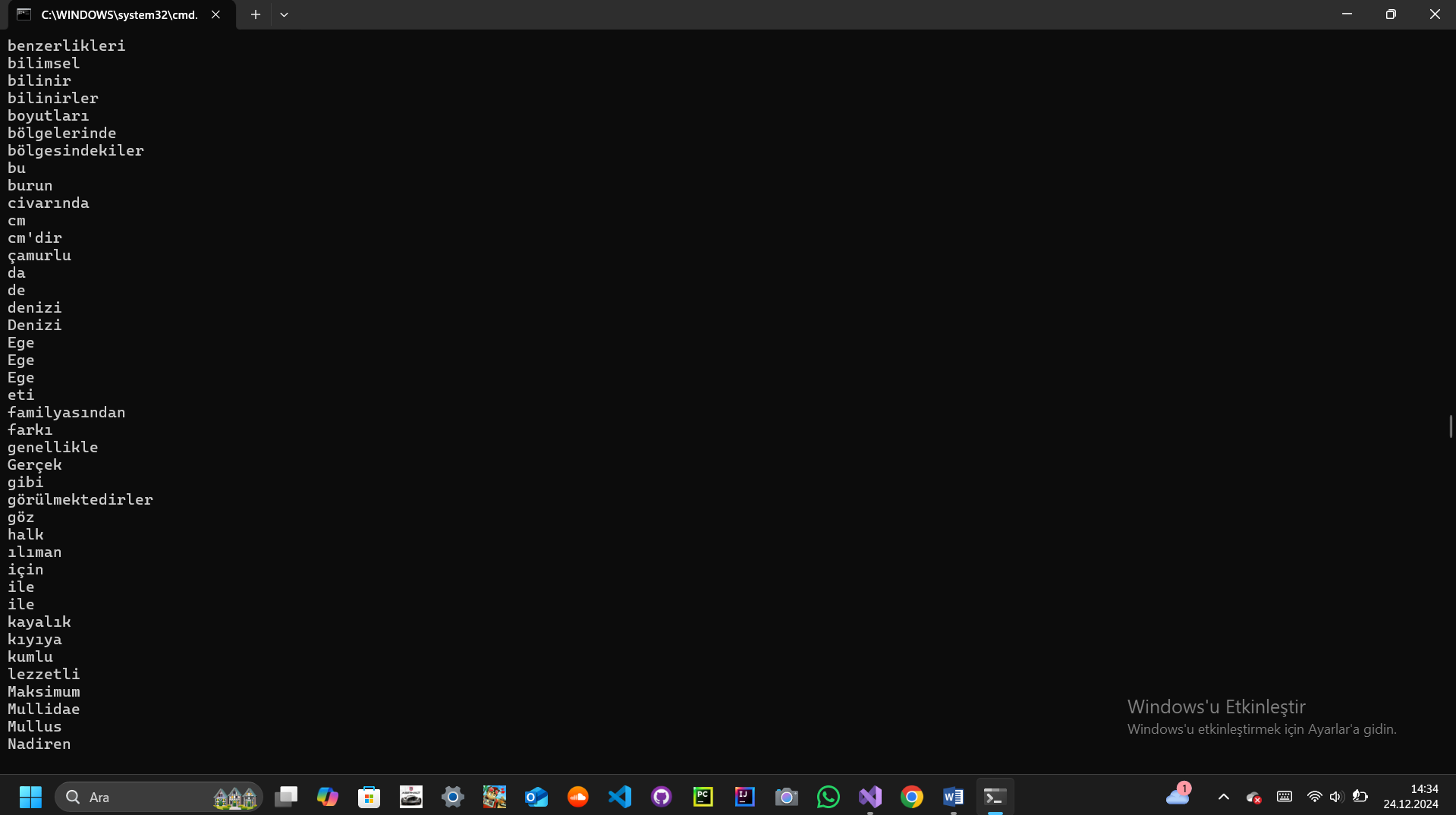
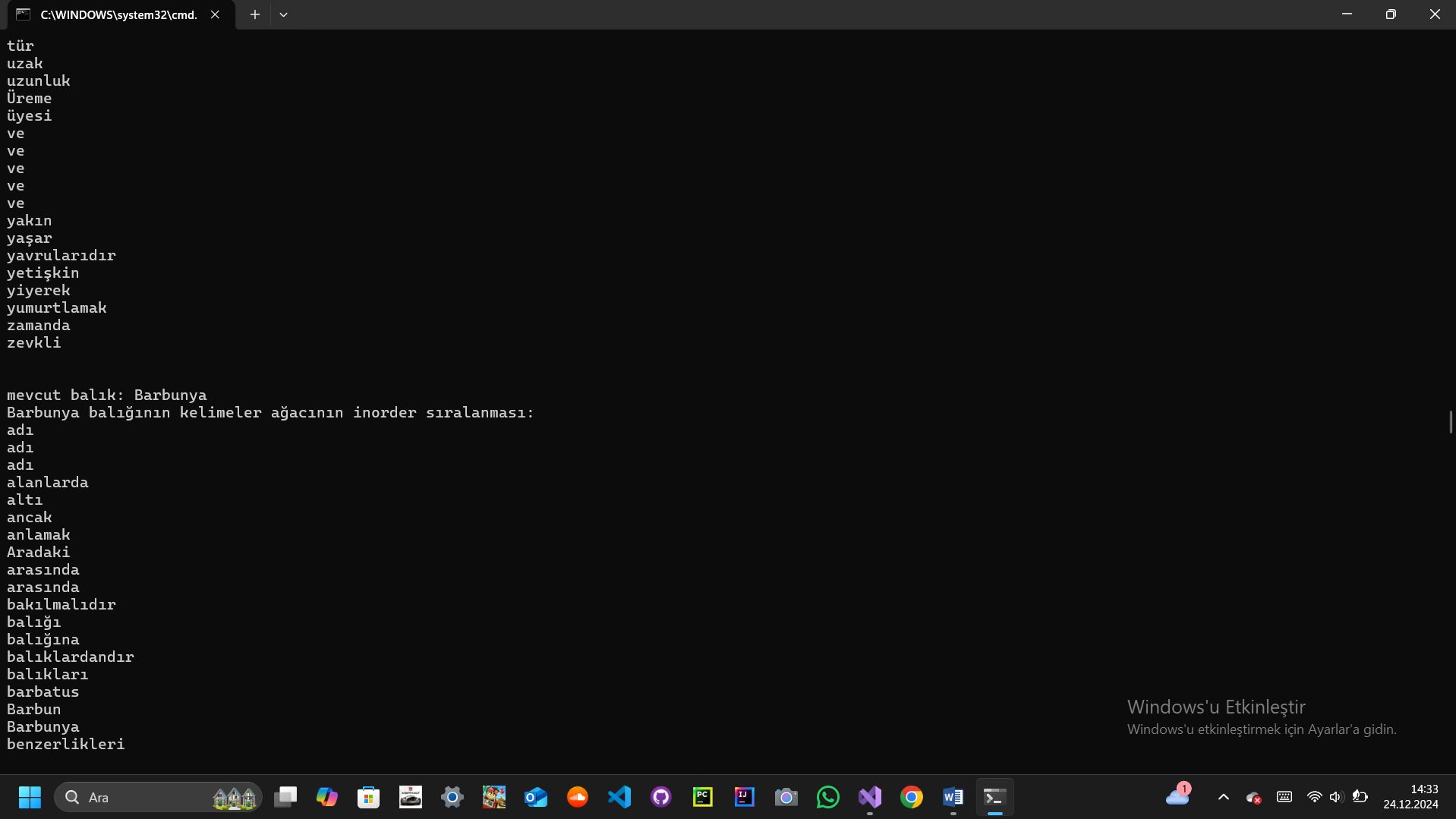
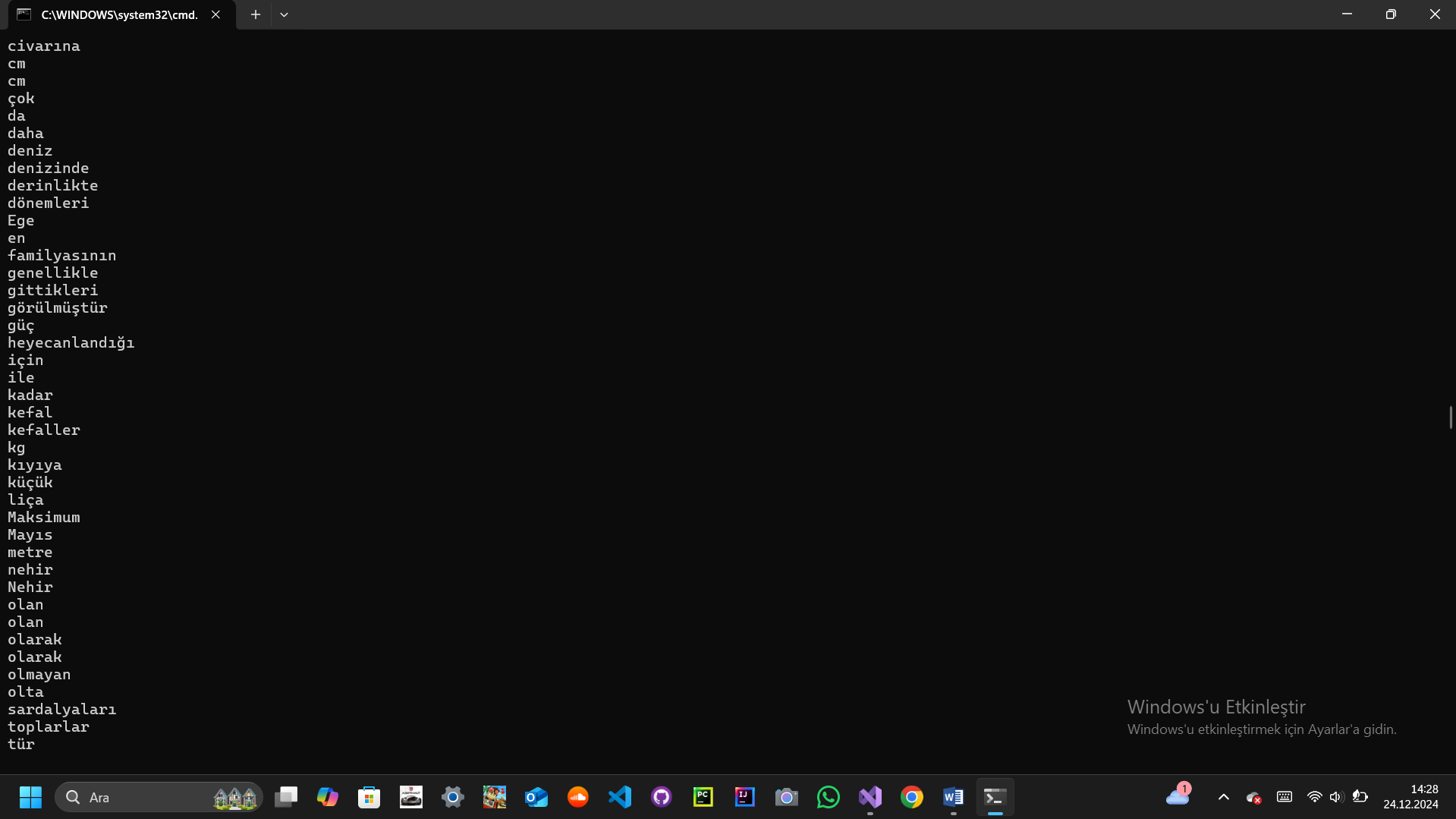
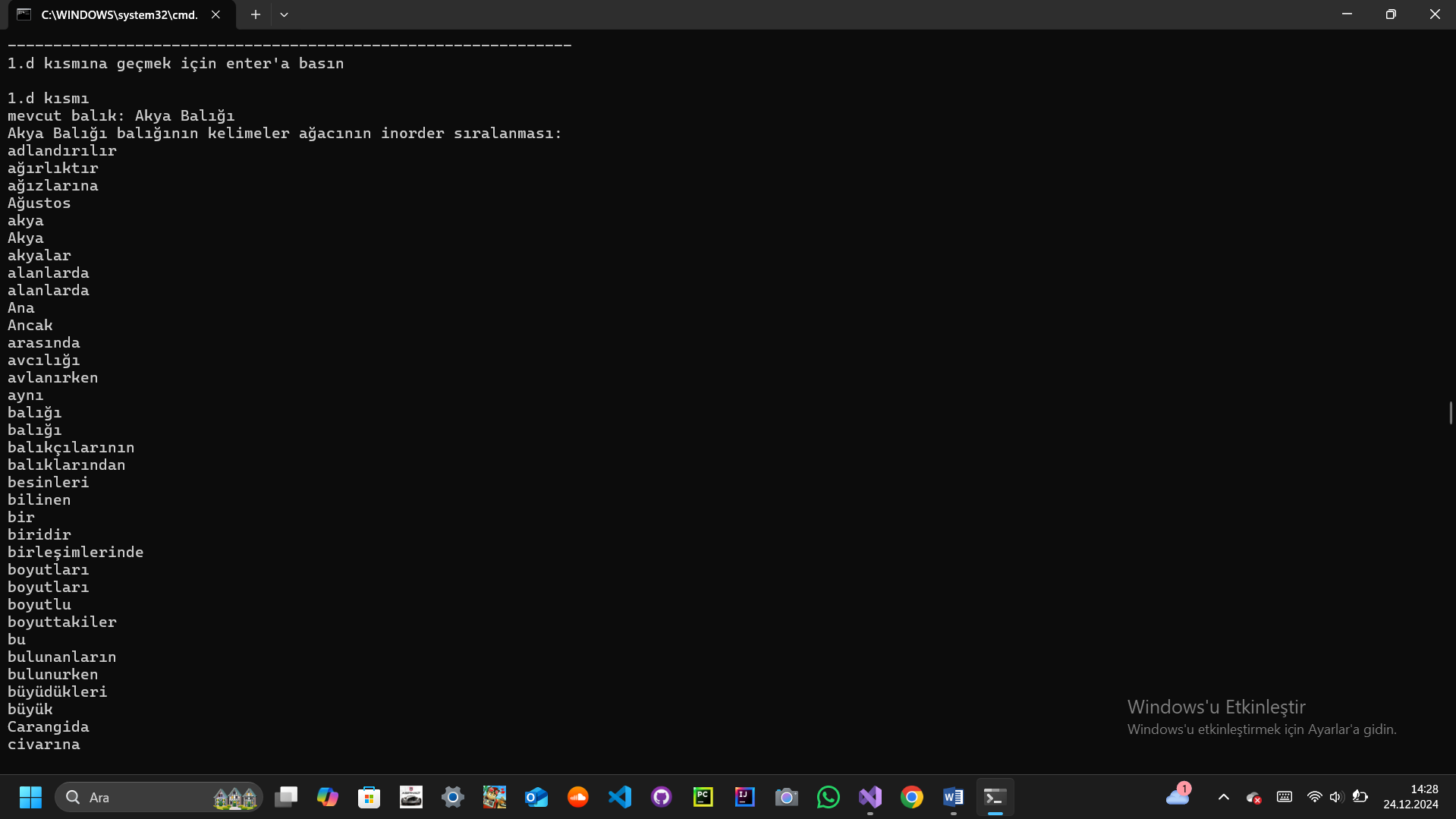


## 1.d Özyineli olarak Dengeli Bir Ağaç Oluşturulması

### 1.d.1 Kodlar

|  |
| --- |
| List<EgeDeniziB> balıkListesi = new List<EgeDeniziB>();  InOrderTraversal2(balıkAgacı.root,balıkListesi);  int sayaç = 0;  foreach (var item in balıkListesi)//balıkList2 dizisi balık adına göre alfabetik olarak dolduruldu  {  balıkList2[sayaç] = item;  sayaç++;  }//balıklist2 dizisi alfabetik olarak oluşturuldu  Console.WriteLine("1.d kısmı"); //1.d kısmı  BalıkAgacı balıkAgacı2 = new BalıkAgacı();//dengeli balık ağacı bu olacaktır  balıkAgacı2.BuildBalancedTree(balıkList2);  balıkAgacı2.inOrder1(balıkAgacı2.root);  Console.WriteLine("balıkağacı2'nin yüksekliği(eleman sayısı 2^5 ve 2^6 arasında olduğu için yüksekliği 5 olmalı): "+balıkAgacı2.height(balıkAgacı2.root));  Console.WriteLine("1.d kısmı tamamlandı");  Console.WriteLine("--------------------------------------------------------------");  Console.WriteLine("2.a kısmına geçmek için enter'a basın");  Console.ReadLine();  public void BuildBalancedTree(EgeDeniziB[] sortedBalıklar)//1.d için  {  this.root = BuildBalancedTreeRecursive(sortedBalıklar,0,sortedBalıklar.Length-1);  }  public Node1 BuildBalancedTreeRecursive(EgeDeniziB[] sortedBalıklar, int startIndex, int endIndex)//1.d içinıı  {  if (startIndex>endIndex)  {  return null;  }  int mid=(startIndex+endIndex)/2;  Node1 node = newNode(sortedBalıklar[mid]);  node.left = BuildBalancedTreeRecursive(sortedBalıklar, startIndex, mid - 1);  node.right=BuildBalancedTreeRecursive(sortedBalıklar, mid + 1, endIndex );  return node;  }  static void InOrderTraversal2(Node1 node,List<EgeDeniziB> values)  {  if (node == null) return;  InOrderTraversal2(node.left,values);  values.Add(node.data);  InOrderTraversal2(node.right,values);  }  public void inOrder1(Node1 rooot)//1.a kısmı burada komple halledildi  {  if (rooot != null)  {  inOrder1(rooot.left);  Console.WriteLine("mevcut balık: "+rooot.data.balıkAdı);  Console.WriteLine(rooot.data.balıkAdı+" balığının bilgi paragrafının alfabetik sıralanması: ");  rooot.data.kelimelerAğacı.inOrder(rooot.data.kelimelerAğacı.root);  Console.WriteLine("\n");  inOrder1(rooot.right);  }    }  public int height(Node1 root)  {  if (root == null)  {  return -1;//yükseklik yok!  }  else  {  int l, r;  l = height(root.left);  r = height(root.right);  return Math.Max(l, r) + 1;  }  } |

### 1.d.2 Ekran görüntüleri



### 1.d.3 Açıklama

Öncelikle elimizde balıkList dizisi vardı(38 tane EgeDeniziB nesnesini içeren), buradaki elemanları inOrderTraversal2() metoduyla alfabetik olarak balıkListesi listesine ekledik oradan da balıkList2 dizisine alfabetik olarak ekledik (bu şekilde soruda bizden istenilen alfabetik balık diziyi elde ettik). Daha sonra BuildBalancedTree metodunda balıkList2 dizisini parametre olarak girip binary şeklinde ağaca elamanları istenildiği gibi ekledik.

# 2) HASH TABLE

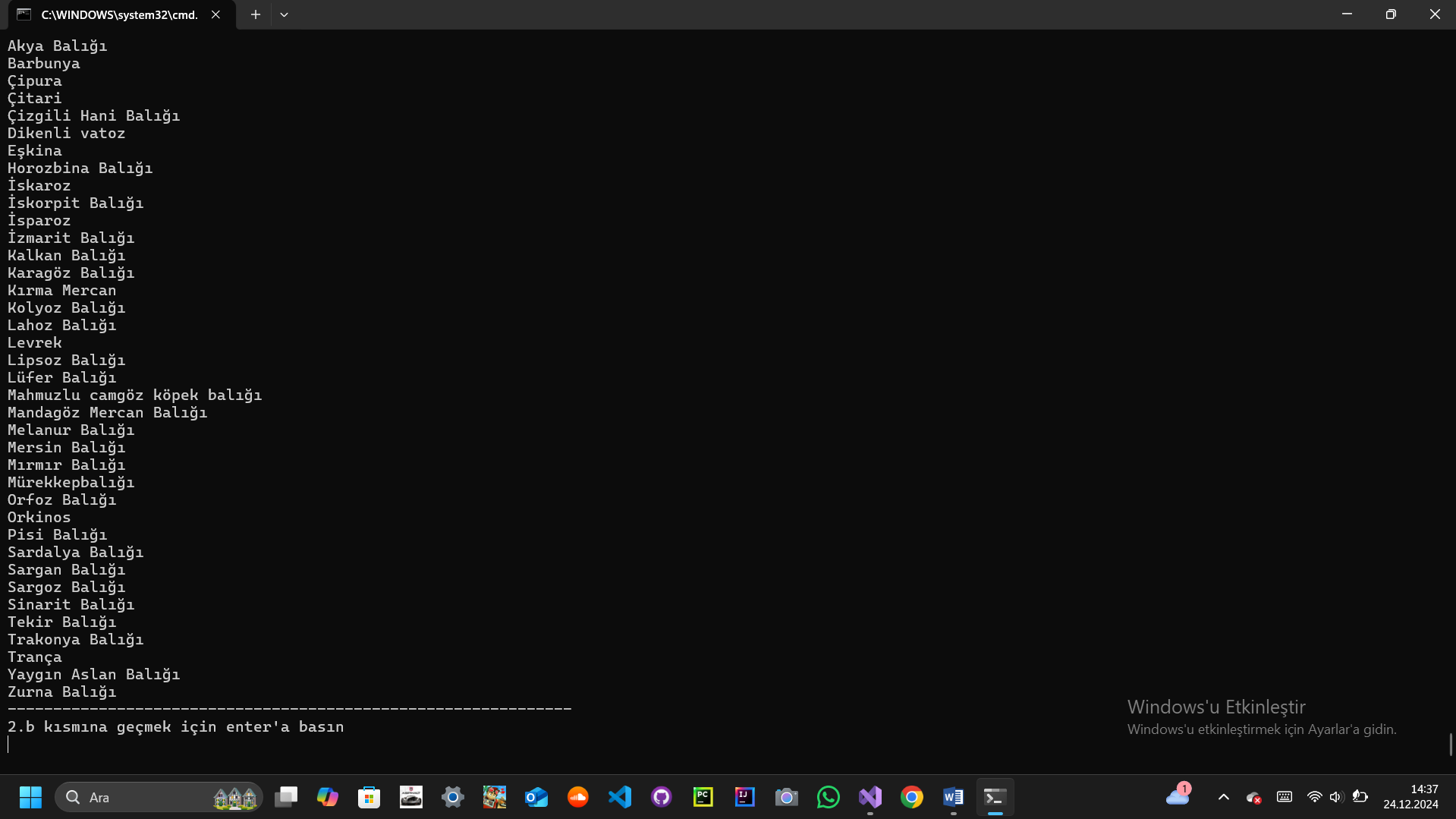
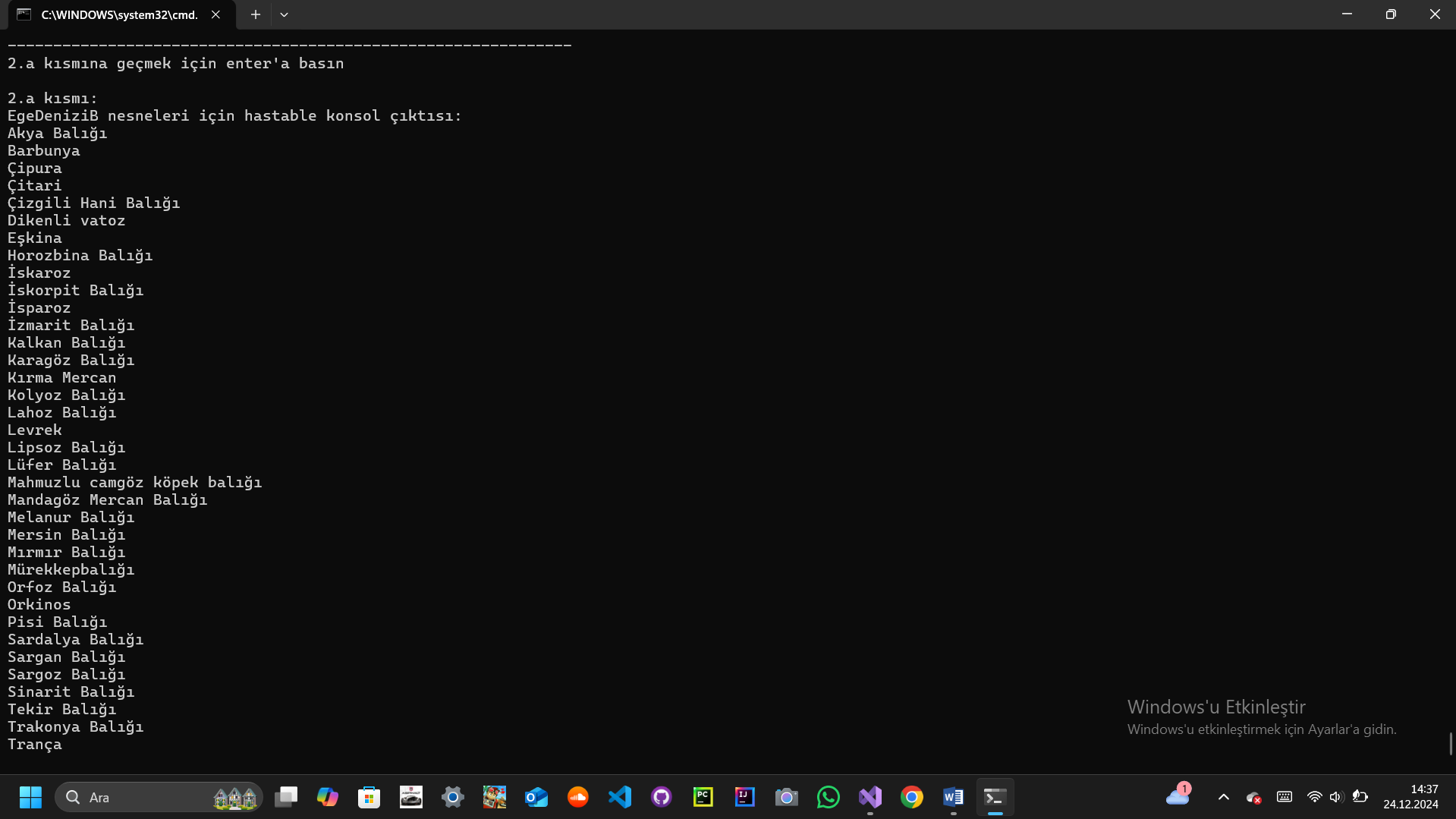
.NET, 4.7.1,C#(windows11)

## 2.a EgeDeniziB Nesnelerini Hash Table’a Yerleştirme

### 2.a.1 Kaynak Kod

|  |
| --- |
| Console.WriteLine("2.a kısmı: ");//2.a ve 2.b kısımları  Hashtable balıkHash = new Hashtable();  for (int i = 0; i < balıkList2.Length; i++)//nesneler alfabetik sırada eklenecek  {  balıkHash.Add(i,balıkList2[i]);  }  Console.WriteLine("EgeDeniziB nesneleri için hastable konsol çıktısı: ");//2.a kısmı  for (int i = 0; i < balıkHash.Count; i++)  {  EgeDeniziB b=(EgeDeniziB)balıkHash[i];  Console.WriteLine(b.balıkAdı);  }//balıkHash initialize işlemi tamamlandı(2.a bitti)  Console.WriteLine("--------------------------------------------------------------");  Console.WriteLine("2.b kısmına geçmek için enter'a basın");  Console.ReadLine(); |

### 2.a.2 Ekran Görüntüleri

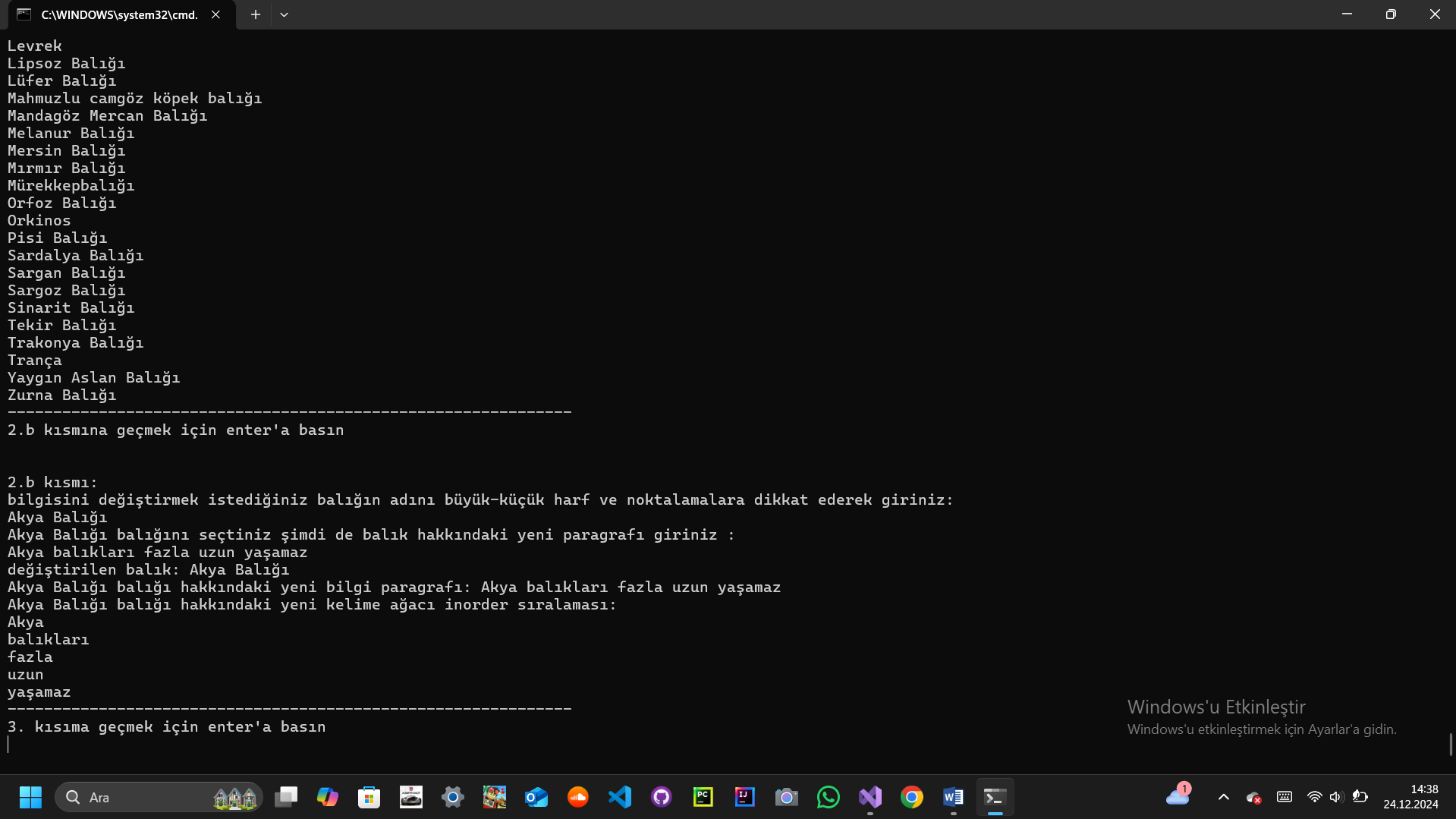


## 2.b Hash Table Güncelleme

### 2.b.1 Kaynak Kod

|  |
| --- |
| Console.WriteLine("\n2.b kısmı: ");//2.b kısmı  Console.WriteLine("bilgisini değiştirmek istediğiniz balığın adını büyük-küçük harf ve noktalamalara dikkat ederek giriniz: ");  string değişecekBalık=Console.ReadLine();  Console.WriteLine(değişecekBalık+" balığını seçtiniz şimdi de balık hakkındaki yeni paragrafı giriniz :");  string yeniParagraf=Console.ReadLine();  EgeDeniziB değişenBalık = new EgeDeniziB();//değişecek balığı tutmak için  int x = 0;//değişecek balığın hashtabledaki key'i için  for (int i = 0; i < balıkHash.Count; i++)  {  EgeDeniziB baluk2 = (EgeDeniziB)balıkHash[i];//hash'daki nesnelere ulaşıyoruz  if (baluk2.balıkAdı.ToLower().Equals(değişecekBalık.ToLower())) {  x = i;  değişenBalık = new EgeDeniziB(baluk2.balıkAdı, yeniParagraf);  balıkHash[i] = değişenBalık;  }  }  EgeDeniziB ba = (EgeDeniziB)balıkHash[x];//güncellenmiş hashtable'da göstermek için bu şekilde yazdım yoksa üstteki for kısmında da direk yazabilirdim bütün bunları.  Console.WriteLine("değiştirilen balık: " + ba.balıkAdı + "\n" + ba.balıkAdı + " balığı hakkındaki yeni bilgi paragrafı: " + ba.bilgi + "\n" + ba.balıkAdı + " balığı hakkındaki yeni kelime ağacı inorder sıralaması: ");  değişenBalık.kelimelerAğacı.inOrder(değişenBalık.kelimelerAğacı.root); //2.b bitti  Console.WriteLine("--------------------------------------------------------------");  Console.WriteLine("3. kısıma geçmek için enter'a basın");  Console.ReadLine(); |

### 2.b.2 Ekran Görüntüsü



### 2.b.3 Açıklama

Önce konsol aracılığıyla kullanıcıdan adını değiştireceğimiz balığı ve balık hakkındaki paragrafı input olarak alıyoruz. Daha sonra for döngüsü ile değiştirilecek balığın indexini alıyoruz, aynı balık adıyla yeni paragrafımızı kullanıp yeni balık nesnesini oluşturup hashtabledaki o index’e atıyoruz. Hashtable güncellendikten sonrada konsola çıktı olarak atıyoruz.

# 3) HEAP VERİ YAPISI (SINIFI)

## 3.a Heap

### 3.a.1 Kaynak Kod

|  |
| --- |
| class MaxHeap  {  private List<EgeDeniziB> heap;  public MaxHeap()  {  heap = new List<EgeDeniziB>();  }  public void Insert(EgeDeniziB egeDeniziB)  {  heap.Add(egeDeniziB);  HeapifyUp(heap.Count - 1);  }  public EgeDeniziB ExtractMax()  {  if (heap.Count == 0)  throw new InvalidOperationException("Heap is empty.");  EgeDeniziB max = heap[0];  heap[0] = heap[heap.Count - 1];  heap.RemoveAt(heap.Count - 1);  HeapifyDown(0);  return max;  }  public int Count => heap.Count;  private void HeapifyUp(int index)  {  while (index > 0)  {  int parentIndex = (index - 1) / 2;  if (string.Compare(heap[index].balıkAdı, heap[parentIndex].balıkAdı) > 0)  {  Swap(index, parentIndex);  index = parentIndex;  }  else  {  break;  }  }  }  private void HeapifyDown(int index)  {  int lastIndex = heap.Count - 1;  while (index <= lastIndex)  {  int leftChildIndex = 2 \* index + 1;  int rightChildIndex = 2 \* index + 2;  int largestIndex = index;  if (leftChildIndex <= lastIndex && string.Compare(heap[leftChildIndex].balıkAdı, heap[largestIndex].balıkAdı) > 0)  {  largestIndex = leftChildIndex;  }  if (rightChildIndex <= lastIndex && string.Compare(heap[rightChildIndex].balıkAdı, heap[largestIndex].balıkAdı) > 0)  {  largestIndex = rightChildIndex;  }  if (largestIndex != index)  {  Swap(index, largestIndex);  index = largestIndex;  }  else  {  break;  }  }  }  private void Swap(int index1, int index2)  {  EgeDeniziB temp = heap[index1];  heap[index1] = heap[index2];  heap[index2] = temp;  }  } |

## 3.b Max Heap’e Yerleştirme

### 3.b.1 Kaynak Kod

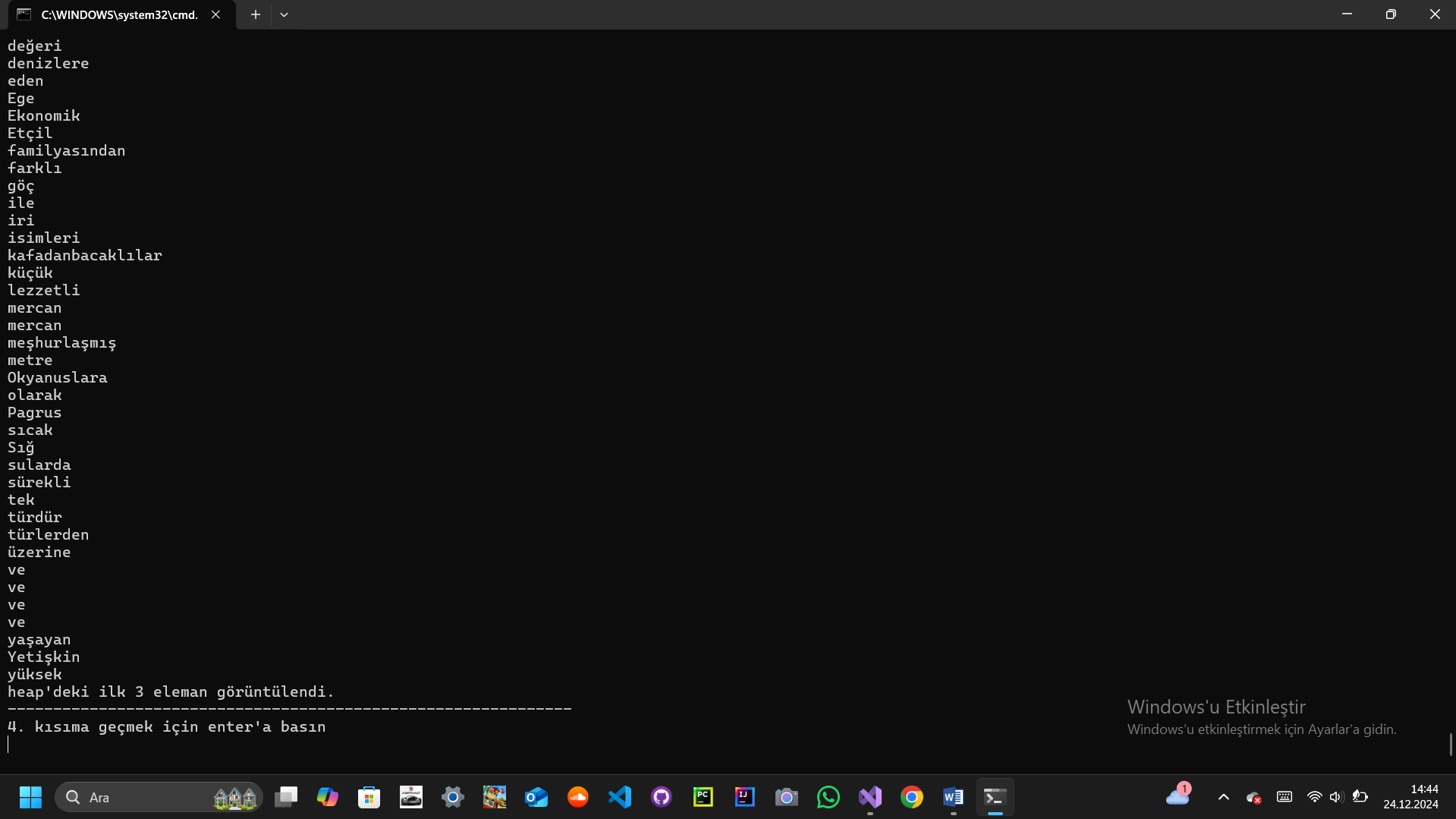
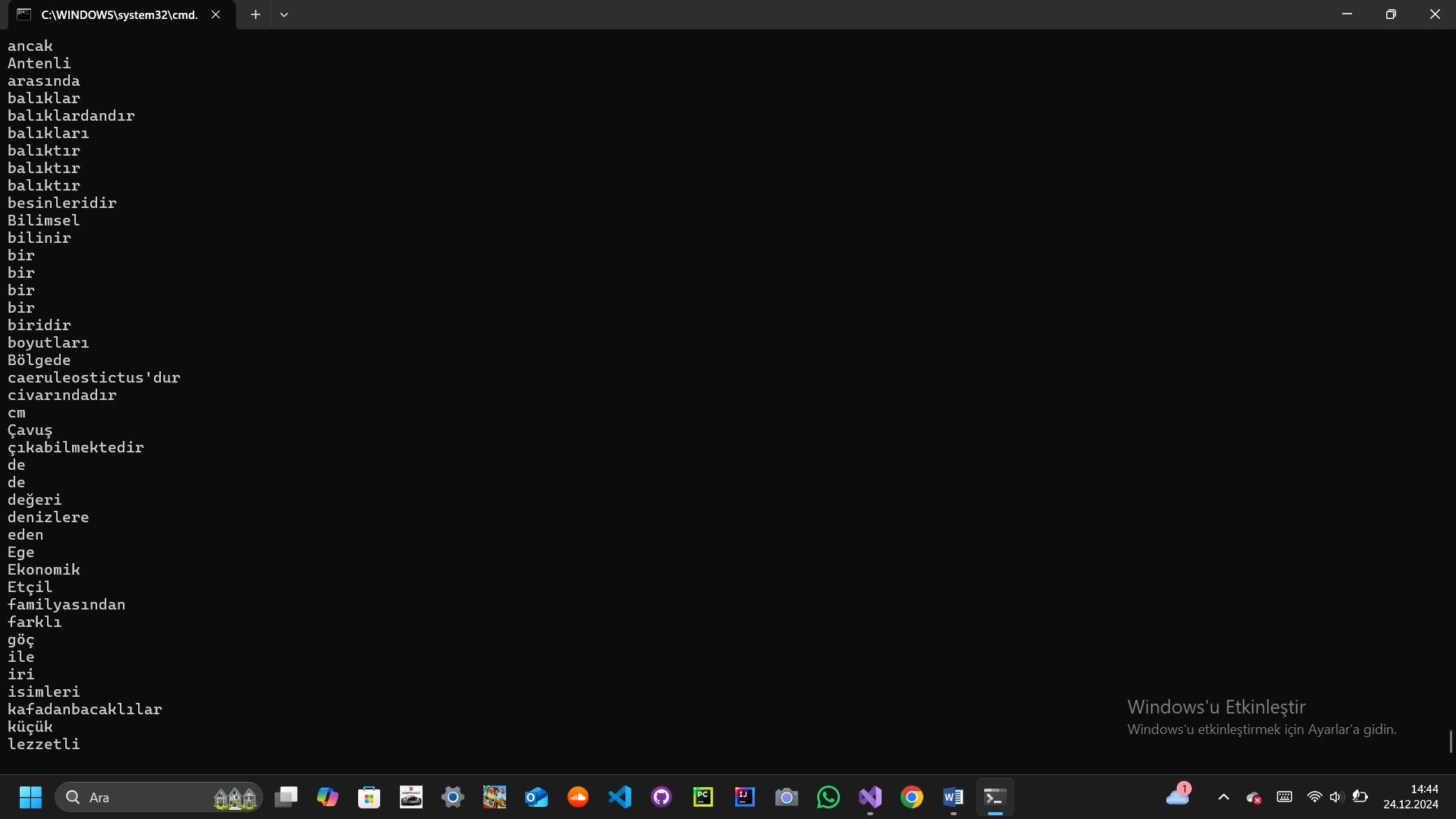
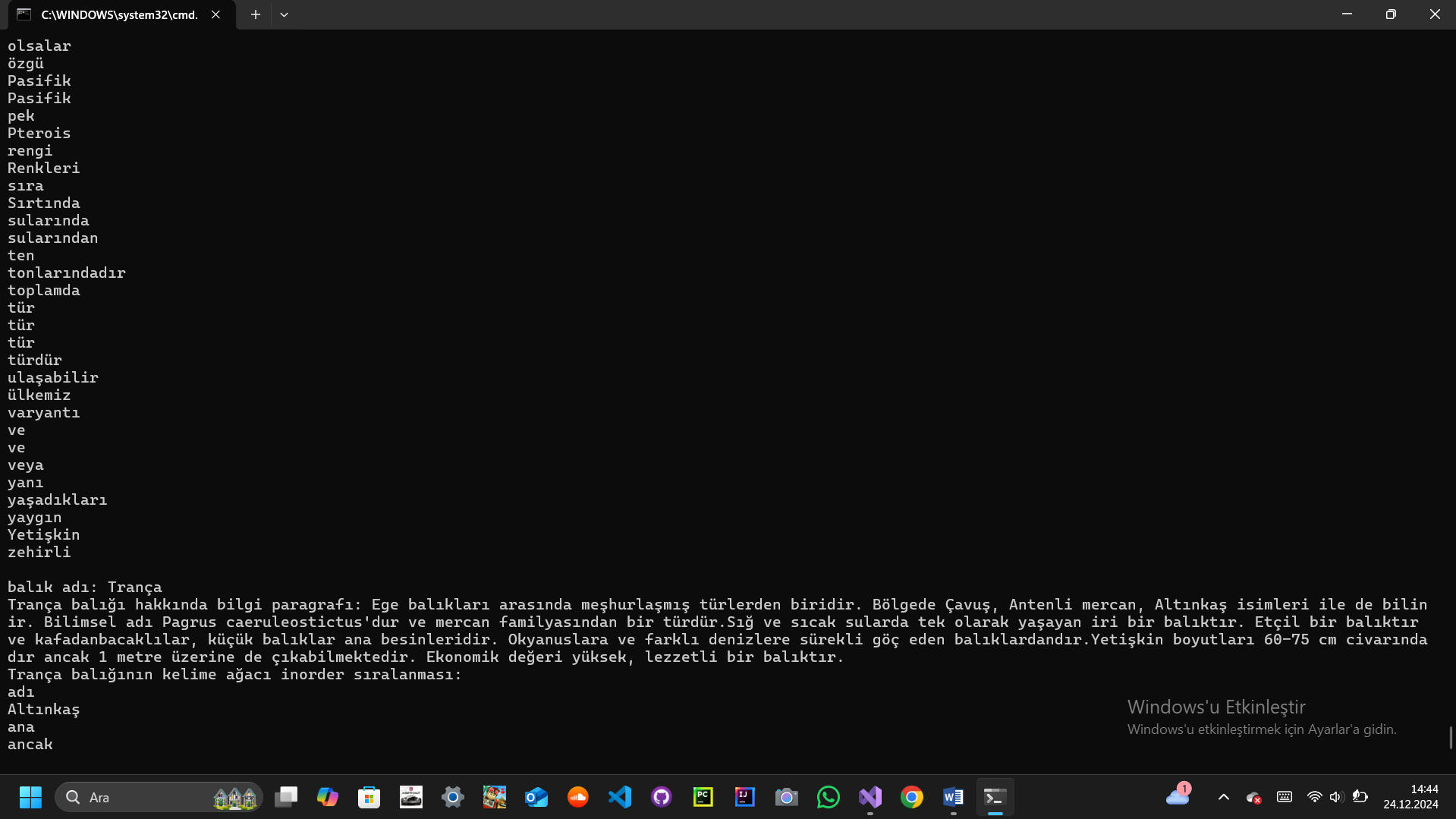
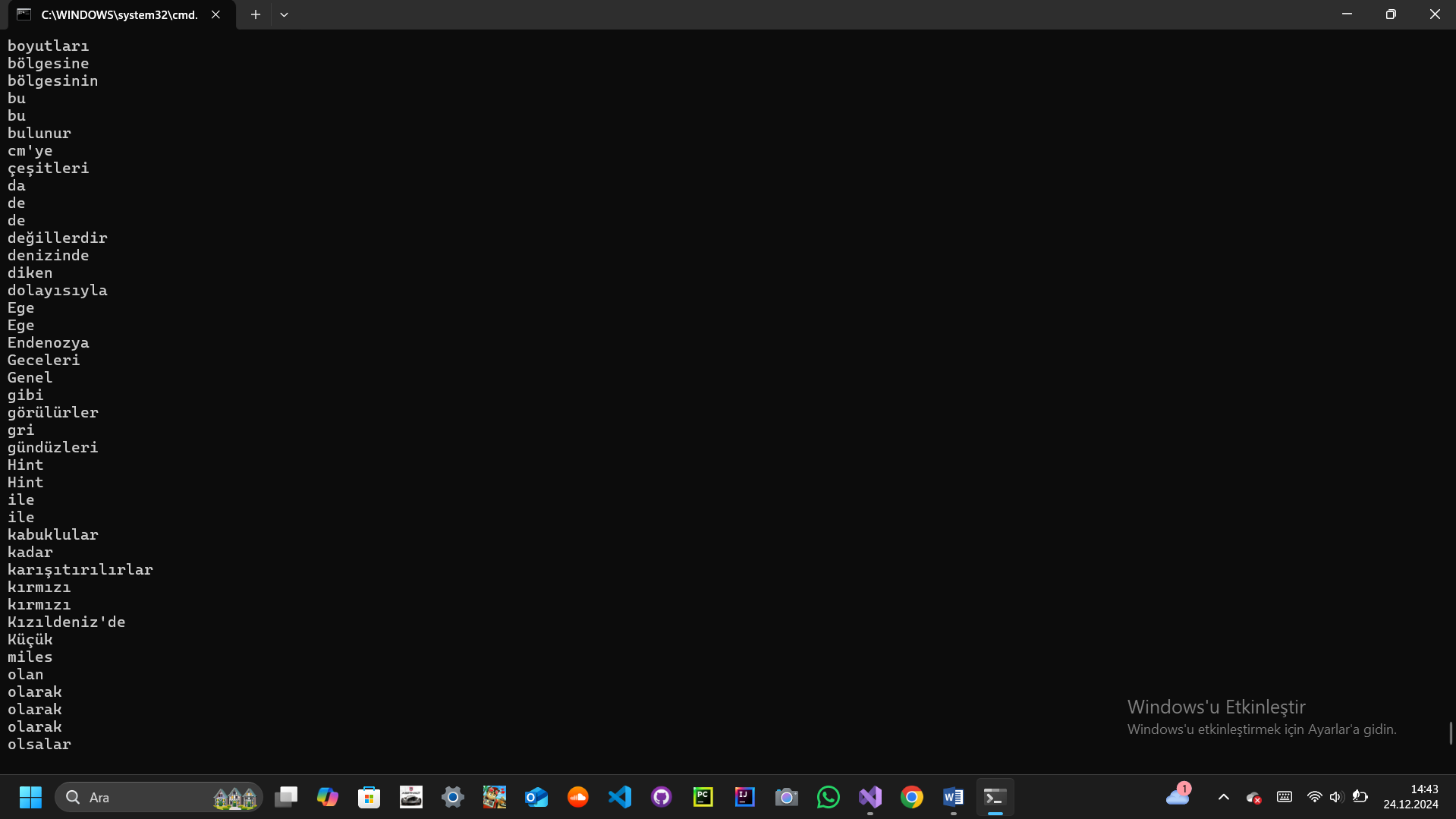
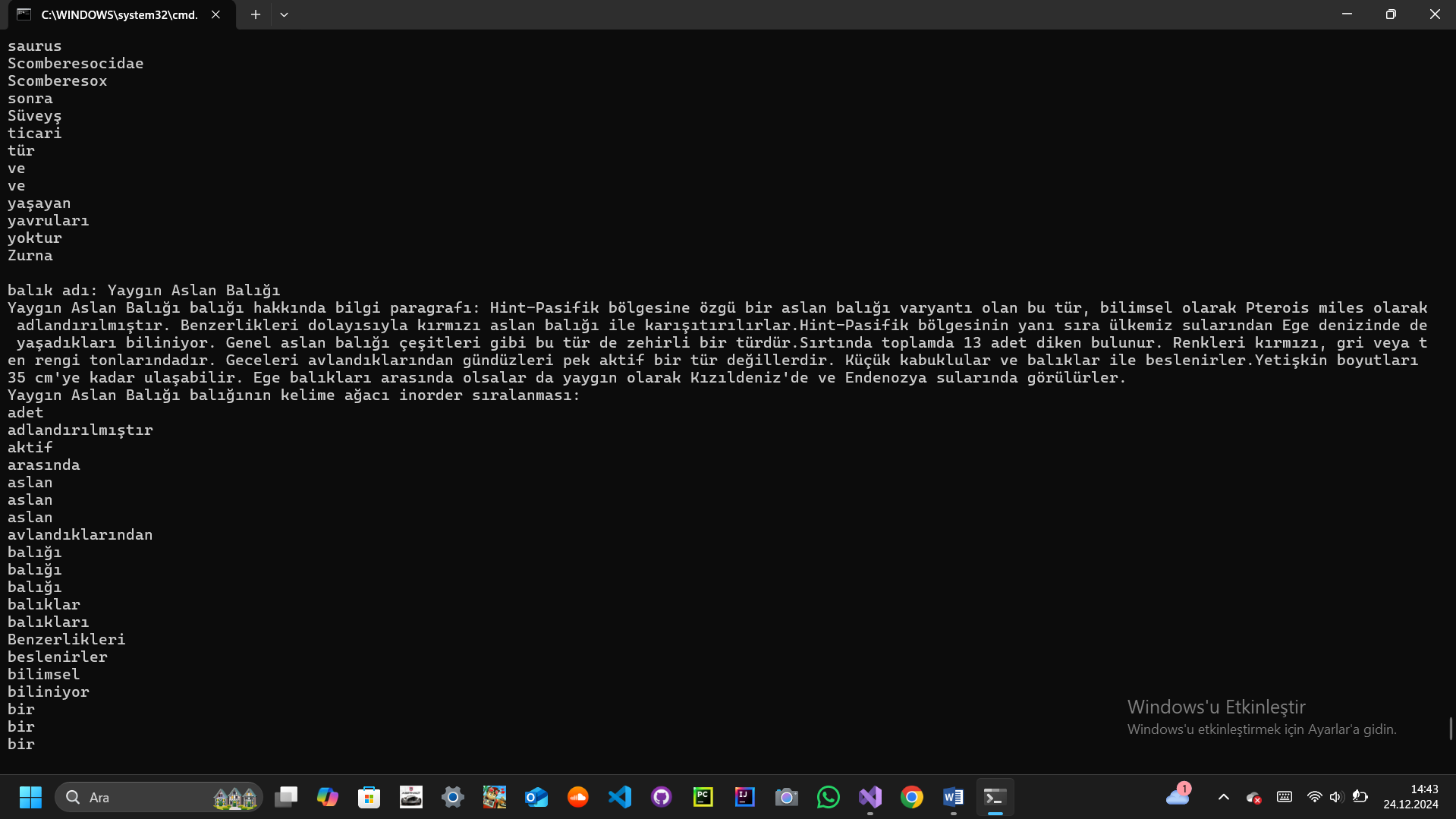
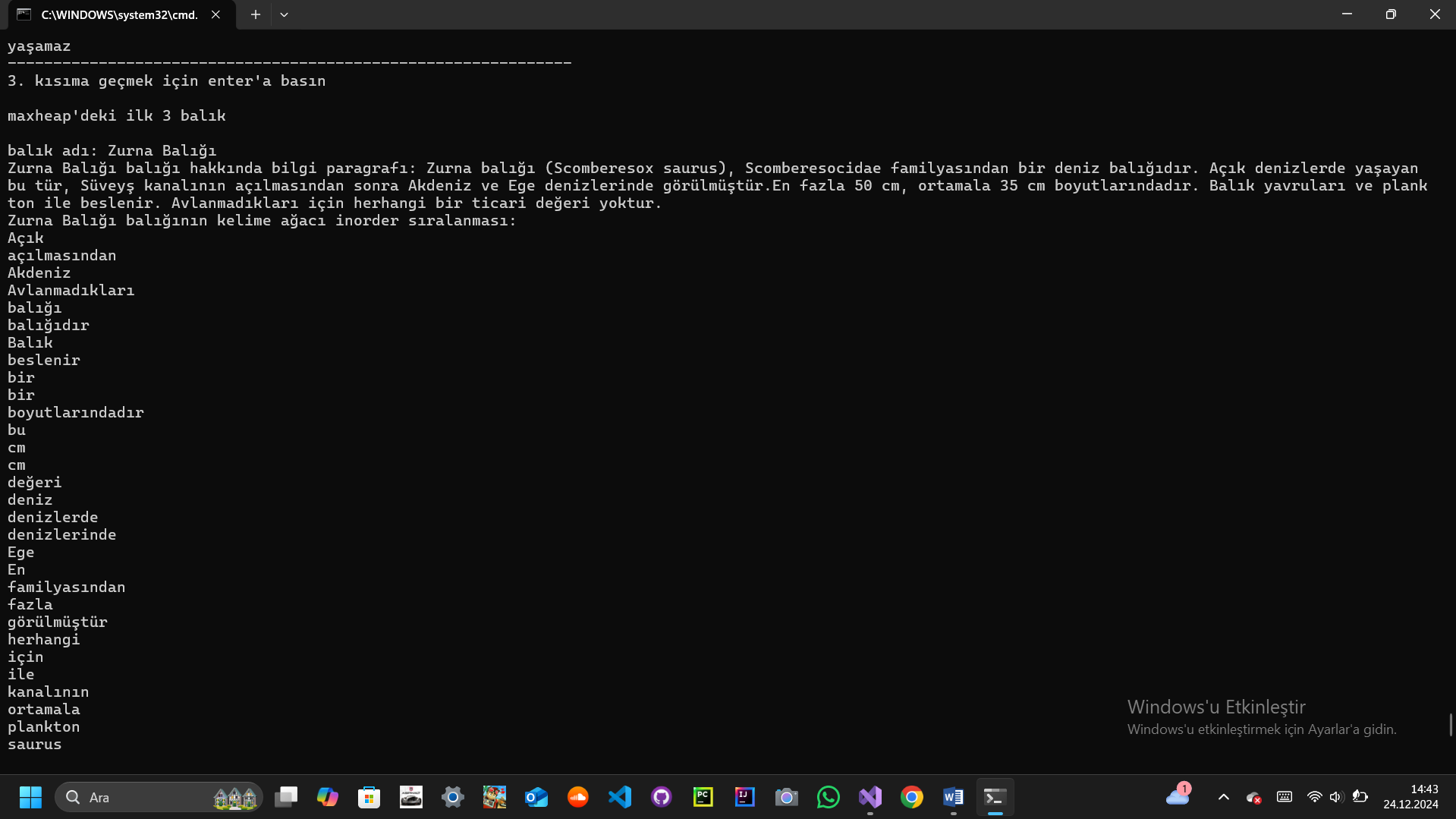
|  |
| --- |
| MaxHeap heapp=new MaxHeap();//3. kısım    for (int i = 0; i < balıkList2.Length; i++)//3.b kısmı  {  heapp.Insert(balıkList2[i]);  }  public void Insert(EgeDeniziB egeDeniziB)  {  heap.Add(egeDeniziB);  HeapifyUp(heap.Count - 1);  } private void HeapifyUp(int index)  {  while (index > 0)  {  int parentIndex = (index - 1) / 2;  if (string.Compare(heap[index].balıkAdı, heap[parentIndex].balıkAdı) > 0)  {  Swap(index, parentIndex);  index = parentIndex;  }  else  {  break;  }  }  } private void Swap(int index1, int index2)  {  EgeDeniziB temp = heap[index1];  heap[index1] = heap[index2];  heap[index2] = temp;  } |

## 3.c Max Heap Listeleme

### 3.c.1 Kaynak Kod

|  |
| --- |
| Console.WriteLine("maxheap'deki ilk 3 balık");  for (int i = 0; i < 3; i++)  {  EgeDeniziB örnekB = heapp.ExtractMax();    Console.WriteLine("\nbalık adı: "+örnekB.balıkAdı);//3.c kısmı  Console.WriteLine(örnekB.balıkAdı+" balığı hakkında bilgi paragrafı: " + örnekB.bilgi);  Console.WriteLine(örnekB.balıkAdı+" balığının kelime ağacı inorder sıralanması: ");  örnekB.kelimelerAğacı.inOrder(örnekB.kelimelerAğacı.root);  }  Console.WriteLine("heap'deki ilk 3 eleman görüntülendi.");  Console.WriteLine("--------------------------------------------------------------");  Console.WriteLine("4. kısıma geçmek için enter'a basın");  Console.ReadLine(); |

### 3.c.2 Ekran Görüntüsü



### 3.c.3 Açıklama

Altyapı olarak EgeDeniziB nesnelerini tutan list yapısı kullandım. Insert()(içindeki parametreyi heap’e(listeye ekleyen) metot) metodunu, HeapifyUp() (eklenen nesneyi heap’de yukarı(adına göre uygun bir yere) taşıyan metot) Swap() (içindeki iki parametrenin yerlerini(indexlerini) değiştiren metot) HeapifyDown()(eklenen nesneyi aşağıya(adına göre uygun bir yere) eklemeye yardımcı metot) metoduyla elemanları maxheap yapısına uygun bir şekilde listeye ekledim.

# 4) SIRALAMA

## 4.a Sıralama Algoritmaları

### 4.a.1 Kaynak Kod

|  |
| --- |
| public static void BubbleSort(int[] array)// 4. kısım: BubbleSort (simple sorting'den bunu seçtim)  {  int n = array.Length;  for (int i = 0; i < n - 1; i++)  {  for (int j = 0; j < n - i - 1; j++)  {  if (array[j] > array[j + 1])  {  // Swap  int temp = array[j];  array[j] = array[j + 1];  array[j + 1] = temp;  }  }  }  }    public static void QuickSort(int[] array, int low, int high)// 4. kısım: QuickSort(advanced sorting için bunu seçtim)  {  if (low < high)  {  int pivotIndex = Partition(array, low, high);  QuickSort(array, low, pivotIndex - 1);  QuickSort(array, pivotIndex + 1, high);  }  }  public static int Partition(int[] array, int low, int high)  {  int pivot = array[high];  int i = (low - 1);  for (int j = low; j < high; j++)  {  if (array[j] < pivot)  {  i++;  // Swap  int temp = array[i];  array[i] = array[j];  array[j] = temp;  }  }  // Swap pivot to correct position  int temp1 = array[i + 1];  array[i + 1] = array[high];  array[high] = temp1;  return i + 1;  } |

### 4.a.2 Açıklama

Bubblesort: Dizi içindeki elemanlar sırayla karşılaştırılır. Daha büyük olan eleman, bir sonraki elemanla yer değiştirir (swap işlemi). Bu işlem, dizideki en büyük elemanın en sona "kabarcık gibi" taşınmasını sağlar. Her iterasyonda en büyük eleman sona sabitlendiği için, sonraki döngülerde daha az eleman kontrol edilir. Her dış döngüde, kontrol edilen eleman sayısı bir azaltılır. Swap işlemi için geçici bir değişken (temp) kullanılır. Genel olarak verilen dizi üzerinde işlem yapıldı, ek bir veri yapısı kullanılmadı(metot üzerinde yaptığımdan dolayı).

Quicksort: Bir "pivot" eleman seçilir (bu örnekte dizinin son elemanı). Dizi, pivot’tan küçük elemanlar ve büyük elemanlar olmak üzere iki alt parçaya ayrılır. Alt parçalar, aynı işlemi tekrar eden şekilde (recursive) sıralanır. Pivot eleman, her adımda doğru yerine yerleştirilir. Pivot seçimi önemlidir; kötü pivot seçimi algoritmayı 𝑂(𝑁^2)’ye düşürebilir. Parçalama işlemi sırasında, elemanlar swap edilerek uygun pozisyonlarına yerleştirilir. Rekürsif çağrılarda başlangıç ve bitiş indekslerinin doğru tutulması gerekir. Yine sadece verilen dizi üzerinde çalışır. Ancak rekürsif çağrılar nedeniyle sistem çağrı yığıtında (call stack) ek bellek kullanımı olur(metodun gereksinimlerinden dolayı).

## 4.b Zaman Karmaşıklığının Hesaplanması

### 4.b.1 Açıklama

// Zaman karmaşıklığı hesabınızı ve diğer sıralama algoritması ile ilgili kıyaslamasını paylaşınız.

Buublesort: En kötü durumda (ters sıralı): 𝑂(𝑁^2)

En iyi durumda (sıralı): 𝑂(𝑁^2)

Avantajı: Basit ve küçük veri setlerinde uygulanabilir.

Dezavantajı: Büyük veri setlerinde çok yavaş.

Quicksort: Ortalama durumda: 𝑂(NlogN)

En kötü durumda (kötü pivot seçimi): 𝑂(𝑁^2)

Avantajı: Büyük veri setlerinde hızlı.

Dezavantajı: Pivot seçimine bağlı olarak kötü performans gösterebilir.

Sonuç: BubbleSort, küçük veri setleri veya öğretici amaçlar için uygun. QuickSort, büyük veri setlerinde ve daha yüksek performans gereksinimlerinde uygun. Zaman karmaşıklığında oluşturduğumuz dizinin ne kadar rastgele olduğu önemli bir noktadır.

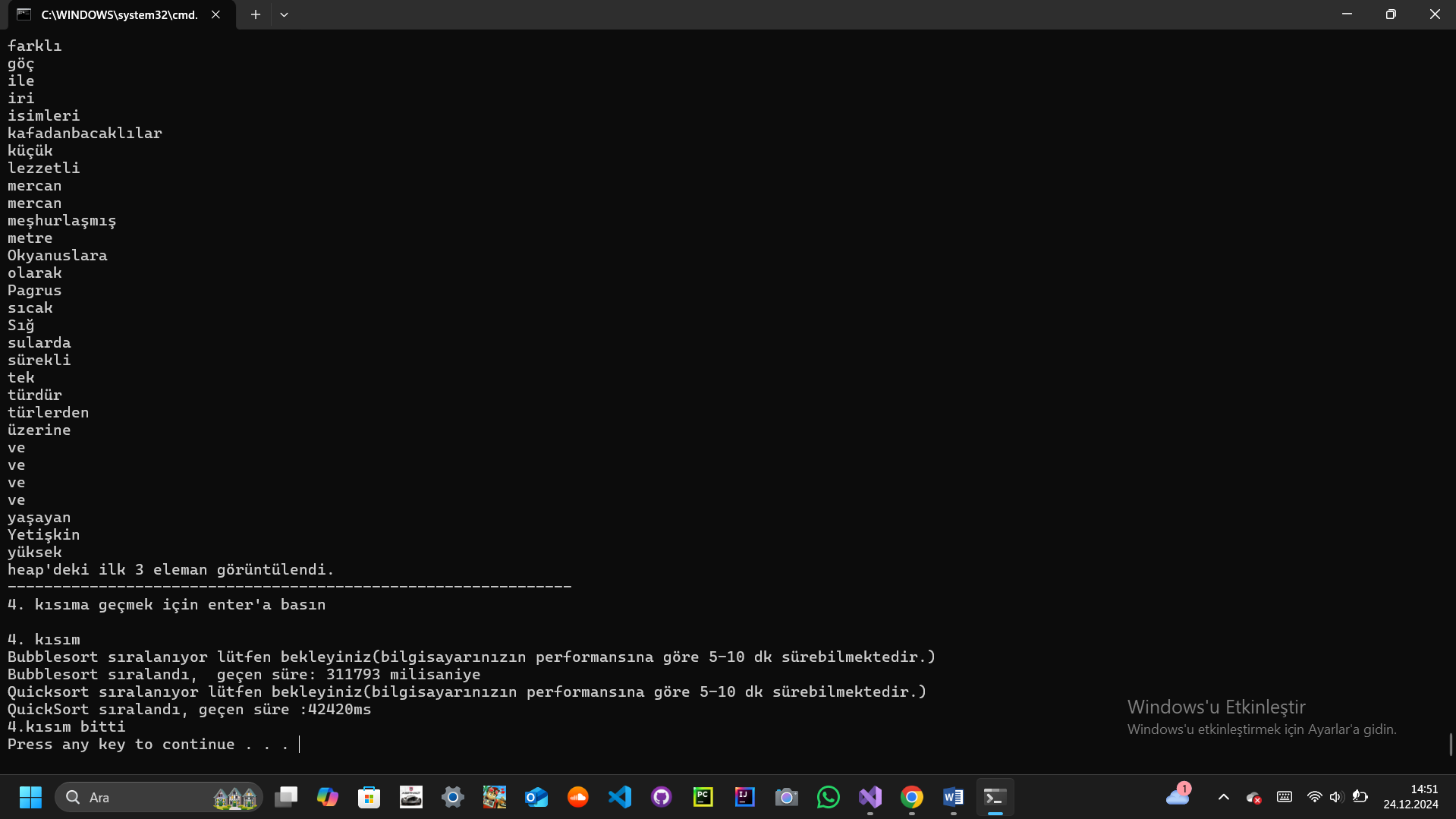
## 4.c Çalışma Zamanının Hesaplanması

### 4.c.1 Kaynak Kod

|  |
| --- |
| Console.WriteLine("4. kısım");  Random random = new Random();  int[] array = new int[100];  for (int i = 0; i < 100; i++)  {  array[i] = random.Next(1, 1000);  }  // Dizi kopyaları  int[] bubbleSortArray = (int[])array.Clone();  int[] quickSortArray = (int[])array.Clone();  // Zaman ölçüm  Stopwatch stopwatch = new Stopwatch();  // BubbleSort zaman ölçüm  Console.WriteLine("Bubblesort sıralanıyor lütfen bekleyiniz(bilgisayarınızın performansına göre 5-10 dk sürebilmektedir.)");  stopwatch.Start();  for (int i = 0; i < 10000000; i++)  {  BubbleSort((int[])bubbleSortArray.Clone());  }  stopwatch.Stop();  Console.WriteLine("Bubblesort sıralandı, geçen süre: "+stopwatch.ElapsedMilliseconds+" milisaniye");  Console.WriteLine("Quicksort sıralanıyor lütfen bekleyiniz(bilgisayarınızın performansına göre 5-10 dk sürebilmektedir.)");  stopwatch.Restart();  for (int i = 0; i < 10000000; i++)  {  QuickSort((int[])quickSortArray.Clone(), 0, quickSortArray.Length - 1);  }  stopwatch.Stop();  Console.WriteLine("QuickSort sıralandı, geçen süre :"+ stopwatch.ElapsedMilliseconds +"ms");  Console.WriteLine("4.kısım bitti"); |

### 4.c.2 Açıklama

Benim bilgisayarımda rastgele 100 elemanlı diziyi 10 milyon kere Buublesort ile sıralamak 311793 milisaniye(yaklaşık 312 saniye), Quicksort ile sıralamak 42420 milisaniye(yaklaşık 42.5 saniye) sürdü. Ancak bu durum oluşturduğumuz dizinin yapısına ve bilgisayarın performansına göre de değişebilmektedir. Aşağıda ekran görüntüsü de bulunmaktadır.



# 5) Görselleştirme

## 5.a Görselleştirmenin Etkisi

### 5.a.1 Açıklama

1) Youtube videosu açıklama: Genel anlamda başlangıç noktalarını veya (bazılarında) orta kısma nasıl ulaştıklarını daha iyi kavradım. Nerede ne kadar karşılaştırma yapıldığı, kaç kere yapıldığı ve totalde ne kadar efektif olduklarını etkili bir şekilde gösterdi.

Log2base2 videoları: Data structures in C# bölümünden 2 tane video izledim:

2)Data structures and it’s real life applications: Bu video veri yapılarının günlük hayattaki uygulamalrını anlatmaktadır.

- 2 boyutlu dizi: bir sinema salonundaki koltuğu 2 boyutlu bir dizi gibi düşünüp koltukları ona göre koordine edebiliriz,

- stack: tarayıcımızda aynı pencerede açtığımız önceki linkler stack şeklinde bellekte tutulur ve geri dönmek istediğimizde bir önceki elemana bu stack yapısı üzerinden ulaşılır(first in last out)

- Queue: bankada sıra numarası almak için makineye gittiğimizde bulunduğumuz sıraya göre bir fiş alırız(first in first out)

- Singly Linked List: Bir şarkı listesi oluşturduğumuzda bir şarkıyı sonrakine bağlar, bu işlem son şarkıya kadar böyle gider(Sonraki şarkıya geçmek için faydalıdır)

- Doubly Linked List: Yine aynı şarkı listesinde bir şarkı önceki şarkıya ve sonraki şarkıya da bağlanabilir(sonraki ve önceki şarkılarda gezinmek için faydalıdır)

- Circular Linked List: Doubly Linked liste benzer, ek olarak son şarkı ve ilk şarkı da birbirine bağlanır; bu şekilde bütün elemanlar birbiri üzerinde bağlanır ve halkamsı bir yapı oluşur.(şarkı listesi sürekli oynatmak istenirse lazım olabilir(replay playlist) veya son şarkıdan ilk şarkıya geçmek istenirse işlevlidir))

- Graphs: Belirli Node’ların birbirlerine bağlanması ile oluşur. Örnek olarak Google Maps verilebilir. Bir ülke düşünelim , bu ülkenin şehirlerini uzaklık değerleri ile beraber Node yapısında tutalım, bir yerden bir yere gitmek istediğimizde Google Maps bu bilgileri kullanarak bize olası yolları gösterebilir (tercihen en kısa yolu)

3) Anatomy of Linked List : Linked list Node adı verilen parçalardan oluşur, Node’un yapısında elde tutulacak asıl veri ve bir sonraki veri için yine Node türünde referans değeri(bellek adresi veya direk adı ile) tutulur. Referans kısmı tercihen Node.next / next şeklinde tutulur.

Örnek: elimizde sayı değeri tutan Node yapısı olsun, int data (field o sınıfa ait özellik demektir.)field’ı sayıyı tutar, Node next field’ı da bir sonrakii değeri tutar. Bu yapı ile ilkin ‘Node head, middle, last;’ şeklinde initialize yaparız, sırayla 10,20,30 değerlerini atarız ; daha sonra ‘head.next=middle; middle.next=last; last.next=null;’ ifadesiyle head’in bir sonraki ögesi middle ögesinin bellek adresine eşitlenir ve aynısı middle için last ile yapılır ancak last ifadesi son ögemiz olduğu için onun next’i boş(null) olarak atanır.

### 5.b Karşılaştırma Yapılması

### 5.b.1 Açıklama

// Öğrenme açısından Videolar ile Etkileşimli Görselleştirme Araçlarını karşılaştırınız.

Videolar ve etkileşimli araçlar veri yapılarının nasıl çalıştığını gayet güzel bir şekilde gösterdi. Ancak ikisinin de biraz daha iyi olduğu bazı hususlar var: etkileşimli araçlar sayesinde veri yapıları üzerinde daha fazla fonksiyona erişme imkanımız oluyor. Mesela linked list etkileşimli aracında linked list için sayılar oluşturup bunları birbirine bağlayabiliriz, listeden eleman çıkarabiliriz ve listeyi komple temizleyebiliriz. Videoda ise bunun sadece ekleme kısmını gördüm,listeyi bizzat oluşturup onlarla etkileşimli bir şekilde işlem yapabilmek de pek tabi sadece izlemekten daha etkili bir öğrenme yöntemi. Videoların artı tarafına gelecek olursam sanırım linked listte bir sonraki elemanın bellek üzerinden erişildiğini göstermesi diyebilirim. Benzer bir durum sıralama algoritmaları tarafı içinde geçerli ancak buradaki farkın o kadarda büyük olduğuna inanmıyorum çünkü etkileşimli aracın tek artı yönü biraz daha yavaş göstermesi ve istediğin birini direk seçebilmen. Özetlemek gerekirse genel anlamda etkileşimli araçları daha faydalı bulduğumu söyleyebilirim.

# 6) Özdeğerlendirme Tablosu

|  |  |  |  |
| --- | --- | --- | --- |
| **Proje 3 Maddeleri** | **Not** | **Tahmini Not** | **Açıklama** |
| **1 a) Ağaç (EgeDeniziB İkili Arama Ağacı)** | **10** | **10** | **EgedeniziB nesnesinde önceki projedeki gibi diğeradı, boyut, bilgi ve ortam fieldları olacakmı tam emin olamadık o yüzden sadece isim, kelimelerAğacı ve bilgi için field açtık(proje için bunlar da yeterli oldu zaten).** |
| **1 b) Derinlik Bulma, Ağacı Listeleme, Düğüm Sayısı Buldurma, Dengeli Ağaç Derinliği Hesaplama** | **10** | **10** | **İstenenleri harfi harfine yerine getirdiğimize inanıyoruz ve tam puan bekliyoruz** |
| **1 c) Arama ve Listeleme** | **5** | **5** | **Arama ve listelemede herhangi bir sıkıntı yaşanmadı, puan kırmak için bir sebep göremiyoruz.** |
| **1 d) Kelime Ağacı Oluşturarak Kelimeleri Sayma** | **5** | **3** | **Dengeli ağaç oluşturuldu ancak tam olarak istenilen yöntemle değil(isterseniz height metodundan da control edebilirsiniz).** |
| **2) Hash Tablosu** | **10** | **10** | **Tablo istenilen şekilde oluşturuldu herhangi bir sıkıntı olmadı.** |
| **3) Yığın Ağacı (Heap)** | **15** | **14** | **Genel olarak istenenleri yapmış olsak da bazı kısımlardan pek de emin olamadık o yüzden bir miktar puan kırılabileceğini düşünüyoruz.** |
| **4) Sıralama Algoritmaları** | **20** | **19** | **Buublesort ve quicksort algoritmalarını seçtik, kitapta class şeklinde yaptı ancak biz metot olarak yaptık belki buradan bir miktar puan kırılabilir.** |
| **5.a) Görselleştirmenin DSA öğrenmeye etkisi** | **5** | **5** | **Anladığımız kadarıyla 3 videoyu da açıkladık, sade ve düzgün bir dille aktardık puan kırılması için bir sebep göremiyorum.** |
| **5.b) Video & Görselleştirme Araçları Karşılaştırma** | **5** | **4** | **Verilen linklerden sadece cs.usfca.edu/ ~galles/visualization/**  **Algorithms/.html adresine baktık, o yüzden açıklama kısmı biraz kısa kalmış olabilir.** |
| **6) Özdeğerlendirme Tablosu** | **10** | **10** | **Özdeğerlendirme tablosunu açık ve nesnel bir şekilde doldurduğumu düşünüyoruz.** |
| **7) İşbölümü (hangi öğrenci hangi maddeleri yaptı ve kaç saat çalıştı?)** | **5** | **5** | **Bütün projeyi kendim yaptım.(Toplam 10 gün sürdü, yaklaşık 50 saat kadar harcadım diyebilirim. 10 gün (30 saat kadar)) zaman harcadım.** |
| **TOPLAM** | **100** | **95** |  |

**Açıklama kısmında yapıldı, yapılmadı bilgisi ve hangi maddelerin nasıl yapıldığı veya neden yapılamadığı kısaca yazılmalıdır. Tahmini not kısmına da ilgili maddeden kaç almayı beklediğinizi yazmalısınız.**

**Not: Raporu teslim edilmeyen projeler değerlendirmeye alınmayacaktır.**