



T.C.
CUMHURİYET ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Kelime Arama Algoritması

HASAN SERT

2015123014

ÖZET:

Kelime Arama Algoritması isminden de anlaşılacağı üzere herhangi bir kelimenin varlığını kontrol etme amacı olan bir projedir. Bu projede aranan bir kelime kelimeler.txt dosyasında bulunuyor mu ? Bulunmuyorsa kelimemizden tek tek harflerin çıkartılarak oluşan yeni kelime dosyada bulunuyor mu ve ya kelimenin komşu harflerinin yer değişmesiyle oluşan yeni kelimenin dosyamızda var olup olmadığı kontrol ediliyor.

Anahtar Kelimeler : Kelimeler Dosyası, Eksilteli Arama ,Komşu Arama ...

GİRİŞ

Kelime Arama Algoritması projesinde bir kelime listesi olmalı. Bu kelimeler proje içerisinden çağırılmalı ve bir diziye atılmalı. Eğer dosyanın yolu yanlış yazılmış ve ya dosya yolu silinmiş ise program bildirim hatası vermeli.

Her bir kelimenin ascii değeri bulunup herhangi bir asal sayıya göre modu alınmalı ve diziye çıkan sonucu indis değeri olarak kabul görüp yerleştirilmeli. Eğer ki farklı iki kelimenin aynı indis değerine denk gelirse quadratik yöntemi ile 1 artırarak dizideki yerine yerleştirilmeli.

Dizi oluşturulduktan sonra arama yapabilmek için kullanıcıdan arama yapılacak kelime istenilmeli. Girilecek kelimeye göre ilk olarak aranacak kelimenin dizide var olup olmadığı kontrol edilmeli eğer ki varsa “Aradığınız kelime kelimeler dizisinde bulunmaktadır” mesajını vermeli. Kelimenin bire bir aynısı yoksa kelimenin tek tek harflerini çıkartarak kelimeler listesinde araması yapılmalı ve kelimenin

komşu karakterlerinin yer değiştirilmesiyle oluşan kelimelerin kelimeler listesinde var olup olmadığı kontrol edilmeli ve ekrana uygun formatta yazdırılmalı.

Proje başlatıldığında Oluşturulan “dizi [indis]: kelime :
ascii ” şeklinde yazdırılmalı. Arama yapıldıktan sonra kullanıcıdan tekrardan arama yapmak istiyor musunuz diye soru döndürülmeli verilen cevaba göre işlemler yapılmalı .

UYGULAMA

```
public static void main(String[] args) {
    listeYaz();
    boolean deger = true;
    Scanner oku = new Scanner(System.in);
    try
    {
        while (deger == true)
        {
            arama();
            System.out.println("Arama yapmak İstiyormusunuz : (True/False)");
            deger = oku.nextBoolean();
        }
    } catch (Exception E) {
        System.out.println("İstek Dışında deger girildi");
    }
}
```

Program çalıştırıldığında ilk listeYaz() fonksiyonu çağırılır. Liste yaz komutu aşağıda verilmiştir.

```
public static void listeYaz() {
    String myList[] = new String[102];
    myList = isimListesi();
    for (int i = 0; i < myList.length; i++) {
        if (myList[i] != null) {
            System.out.println "[" + i + "]" + ":" + myList[i] + " " + ascii(myList[i]);
        }
    }
}
```

listeYaz() methodu program çalıştırıldığında ekranda kelimelerin ve ascii karşılıklarının yazdırılacağı kod bölümüdür.

ascii() methodu aşağıda verilmiştir.

```
public static int ascii(String satir)
{
    String cevir = satir.toLowerCase();
    int kelime_toplam = 0;
    for (int a = 0; a < cevir.length(); a++)
    {
        char c_karakter = cevir.charAt(a);
        int ascii_no = (int) c_karakter;
        kelime_toplam += ascii_no * Math.pow((a + 1), 4);
    }
    return kelime_toplam;
}
```

ascii() methodunda gelen kelimenin tek tek harflerinin ascii karşılıkları alınıp belirli işlemlerden sonra kelime_toplam değişkenine deger aktarılıyor ve geri int türünde kelime_toplam döndürülüyor.

Main bloğundaki sonraki işlem arama() methodudur. Arama methodu aşağıda verilmiştir.

```

public static void arama() //Arama yapacağımız method
{
    int[] secili_dizi = new int[102];
    String myList[] = new String[102];

    myList = isimListesi();
    secili_dizi = asciiListesi();
    ArrayList array = new ArrayList();
    System.out.println("*****ARAMA BOLUMU*****");
    System.out.print("Aranacak adi giriniz : ");
    Scanner input = new Scanner(System.in);
    String metin;
    metin = input.next();
    int kelime_toplam = 0;
    String cevir = metin.toLowerCase();
    String dizi[] = new String[cevir.length()];
    for (int i = 0; i < myList.length; i++) {
        array.add(myList[i]);
    }
    kelime_toplam = ascii(cevir);
    String[] eksiltili_dizi = new String[cevir.length()];
    String[] komsu_dizi = new String[cevir.length() - 1];
    boolean deger = false;
    for (int i = 0; i < secili_dizi.length; i++) {
        if (secili_dizi[i] == kelime_toplam)
        {
            System.out.println(" Girdiginiz *" + metin + "*" kelimesi metin dosyasinda bulunmaktadır...");
            deger = true;
        }
    }
}

```

```

int sayac = 0;
if (deger == false) {
    for (int i = 0; i < secili_dizi.length; i++) {
        eksiltili_dizi = eksiltili_arama(cevir);
        komsu_dizi = komsu_arama(cevir);
        for (int j = 0; j < eksiltili_dizi.length; j++) {
            if (ascii(eksiltili_dizi[j]) == secili_dizi[i]) {
                System.out.println("Girdiğiniz *" + metin + "*" kelimesi metin dosyasında *" + eksiltili_dizi[j] + "*" olarak bulunmuştur");
                sayac++; //Sayac kullanmamızın sebebi kelimeni listede hiç bulunamama durumunda kontrol sayacı olarak görev almıştır
                break;
            }
        }
        for (int j = 0; j < komsu_dizi.length; j++) {
            if (ascii(komsu_dizi[j]) == secili_dizi[i]) {
                System.out.println("Girdiğiniz *" + metin + "*" kelimesi metin dosyasında *" + komsu_dizi[j] + "*" olarak bulunmuştur");
                sayac++; //Yukarıdaki sayac ile aynı.Bu sayac eğer 0'da kalmassa demekki hiç bu if lere girmemiştir demektir
                break; //İşlemi sonlandırma kısmı sayaca gerek kalmıyor gibi gözüksede olmadığında hata veriyor
            }
        }
    }
    if (sayac == 0) {
        System.out.println("**" + metin + "*" ile alakalı metin dosyasında benzer kelime bulunamamıştır ..."); //sayacın 0 ve kelimenin listede
    }
    sayac = 0;
}
//Yerleştirdiğimiz kelimelerin indisi,kelime adı,ascii karşılıklarını ekrana yazdırma işlemi
}

```

Arama() fonksiyonu çağırıldığında 2 tane dizi oluşturuluyor .

Sonra asciiListesi oluşturulan seçili_dizi ye aktarılıyor.

asciiListesi aşağıda verilmiştir.

```
public static int[] asciiListesi()
{
    int[] secili_dizi = new int[102];
    String[] dizi = new String[dosya_okuma().length];
    dizi = dosya_okuma();
    int kelime_toplam = 0;
    for (int i = 0; i < dizi.length; i++)
    {
        kelime_toplam = ascii(dizi[i]);
        for (int j = 0; j < secili_dizi.length; j++)
        {
            int mod = kelime_toplam % 101;
            if (secili_dizi[mod] == 0) {
                secili_dizi[mod] = kelime_toplam;
                break;
            } else {
                int kalan = (mod + (j * j)) % 101;
                if (secili_dizi[kalan] == 0) {
                    secili_dizi[kalan] = kelime_toplam;
                    break;
                }
            }
        }
        kelime_toplam = 0;
    }
    return secili_dizi;
}
```

AsciiListesi methodunda yeni bir dizi oluşturuluyor.

Sonrasında dosya_okuma fonksiyonu çağırılıyor.

dosya_okuma() fonksiyonu aşağıda verilmiştir.


```

public static String[] dosya_okuma()
{
    ArrayList<String> arraylist = new ArrayList<String>();
    try {
        FileReader fr = new FileReader(path);
        BufferedReader br = new BufferedReader(fr);
        String satir;
        int sayac = 0;
        while ((satir = br.readLine()) != null)
        {
            String cevir = satir.toLowerCase();
            arraylist.add(cevir);
            sayac++;
        }
        br.close();
    } catch (Exception E)
    {
        System.out.println("Dosya bulanamadi ...");
    }
    String liste[] = new String[arraylist.size()];
    for (int i = 0; i < arraylist.size(); i++) {
        liste[i] = arraylist.get(i);
    }

    return liste;
}

```

Burada belirtilen yolda verilen dosya çekilir dosya bulunmuyorsa gereken hata mesajı verilir.

arama() methodu devamında ;

1.Kelimelerin asciilerinin bulunduğu dizi

2.Kelimelerin bulunduğu dizi.İlk olarak aranan kelimenin ascii karşılığı dizide var mı diye kontrol ediliyor kelime listede bulunuyorsa “Girdiğiniz kelime metin dosyasında bulunmaktadır”diye mesaj

veriyor .Eğer ki listede kelime bu şekilde bulunmuyorsa kelime eksiltili_arama() ve komsu_arama() fonksiyonlarına yönlendiriliyor.Eksiltili_arama() ve komsu_arama() aşağıda verilmiştir.

```
public static String[] eksiltili_arama(String satir)
{
    String kelime = "";
    String dizi[] = new String[satir.length()];
    int sayac = satir.length() - 1;
    while (sayac >= 0)
    {
        for (int i = 0; i < satir.length(); i++)
        {
            if (i != sayac) {
                kelime += satir.charAt(i);
            }
        }
        dizi[sayac] = kelime;
        kelime = "";
        sayac--;
    }
    return dizi;
}
```

Burda gelen kelimedden her harfin sıra ile çıkması ile oluşan kelimeler dizisi geri döndürülüyor .

```

public static String[] komsu_arama(String satir)
{
    String kelime = "";
    String dizi[] = new String[satir.length() - 1];
    char karakterDizi[] = new char[satir.length()];
    int sayac = satir.length() - 1;
    char degisken;
    while (sayac > 0) {
        for (int k = 0; k < karakterDizi.length; k++) {
            karakterDizi[k] = satir.charAt(k);
        }
        degisken = karakterDizi[sayac];
        karakterDizi[sayac] = karakterDizi[sayac - 1];
        karakterDizi[sayac - 1] = degisken;
        for (int j = 0; j < satir.length(); j++) {
            kelime += karakterDizi[j];
        }
        dizi[sayac - 1] = kelime;
        kelime = "";
        sayac--;
    }
    return dizi;
}

```

Burda ise gelen kelimenin komşu karakterlerinin yer değişmesiyle oluşan kelime listesi geri döndürülüyor.

arama() fonksiyonunda bu methodlar çağırıldıktan sonra tekrardan listede bu kelimelerin varlığı kontrol ettirilir eğer ki bulunuyorsa “Arama yaptığınız kelime eksiltili arama yapıldığında bulunmuştur” ve ya “Arama yaptığınız kelime komsu arama sonucunda bulunmuştur” şeklinde ekrana yazdırılıyor. Bunlar denendikten sonra hiçbir şekilde geri dönüş sağlanamadıysa “Kelime hiçbir

şekilde dosyada bulunmuyor mesajı veriyor” En sonun da arama() methodunda da son işlem gerçekleştikten sonra main bloğundan devam edilir ve kullanıcıdan tekrardan arama yapmak istediği sorulur verilen cevaba göre işlemler gerçekleştiriliyor.

SONUÇLAR

[41]:afra 35795	run:	[81]:dicle 100572
[42]:amelya 239007	[0]:asuela 230482	[82]:hasan 104553
[43]:ekdi 36793	[1]:alisa 100395	[83]:fubtol 248951
[44]:dilberay 974998	[2]:ceyla 99789	[84]:ekin 38478
[45]:derya 102551	[3]:elmas 107365	[85]:ayse 37204
[46]:belemir 514237	[4]:arin 38586	[86]:bedran 242183
[47]:ked 9823	[5]:ecem 37770	[87]:anita 100683
[48]:edi 10206	[6]:anka 35356	[88]:cansel 243094
[49]:ceylin 247349	[7]:eftalya 493174	[89]:canel 103917
[50]:belinda 468589	[8]:anise 102927	[90]:elif 36446
[51]:ceylan 242349	[9]:berran 243317	[91]:dilay 110985
[52]:ebru 40855	[10]:beltun 255843	[92]:cansın 373186
[53]:dilek 103259	[11]:belma 98991	[93]:arya 36554
[54]:bade 35606	[12]:aykal 103032	[94]:altın 106851
[55]:adel 37526	[13]:anar 38898	[95]:ela 9686
[56]:ahlem 104490	[14]:ceren 105555	[96]:benice 230275
[57]:berceste 940669	[15]:utbol 105827	[97]:abendam 487927
[58]:dilhan 240337	[16]:alpike 235548	[98]:destan 243912
[59]:zidan 103484	[17]:destan 243912	[99]:adelya 238863
[60]:ceylinaz 979958	[18]:eser 39306	[100]:behrem 243711
[61]:defne 101263	[19]:esmeray 524109	
[62]:almila 230746	[20]:cannur 259590	
[63]:dilruba 472742	[21]:alvina 232725	
[64]:eda 9558	[22]:amara 99507	
[65]:duygu 111266	[23]:demet 108901	
[66]:zimb 35719	[24]:deste 103852	
[67]:aysar 107430	[25]:ayren 105873	
[68]:ahmet 108946	[26]:bediz 112944	
[69]:benli 103897	[27]:pınar 106784	
[70]:diler 107634	[28]:esin 38606	
[72]:dolunay 525559	[29]:derin 106580	
[73]:ece 9866	[30]:deniz 113756	
[74]:berva 101781	[31]:dilara 232322	
[75]:afitap 245707	[32]:diclehan 918813	
[76]:adile 100975	[33]:didem 103861	
[77]:damla 98754	[34]:futbo 105833	
[78]:ceyda 97741	[35]:berinay 522811	
[79]:hazla 99811	[36]:berre 103257	
[80]:dilan 104110	[37]:cilvenaz 979029	
	[38]:ahenk 104977	
	[39]:kedi 36703	
	[40]:besra 100838	

```
*****ARAMA BOLUMU*****
```

```
Aranacak adi giriniz :
```

Aranacak ad girilmesi isteniyor bir kelime girelim...

```
*****ARAMA BOLUMU*****
```

```
Aranacak adi giriniz : kedi|
```

Sonuç ...

```
Aranacak adi giriniz : kedi
```

```
Girdiginiz *kedi* kelimesi metin dosyasinda bulunmaktadir...
```

```
Arama yapmak İstiyormusunuz : (True/False)
```

```
|
```

Tekrardan arama yapalım ...

```
*****ARAMA BOLUMU*****
Aranacak adi giriniz : kedi
Girdiginiz *kedi* kelimesi metin dosyasinda bulunmaktadir...
Arama yapmak istiyormusunuz : (True/False)
True
*****ARAMA BOLUMU*****
Aranacak adi giriniz : keda
Girdiginiz *keda* kelimesi metin dosyasinda *ked* olarak bulunmuştur
Girdiginiz *keda* kelimesi metin dosyasinda *eda* olarak bulunmuştur
Arama yapmak istiyormusunuz : (True/False)
|
```

Yapılan arama sonuçları ekrana yansıtılmıştır.

KAYNAKLAR

```
package muhendislikpro;

import java.io.BufferedReader;

import java.io.FileReader;

import java.util.ArrayList;

import java.util.Scanner;

public class MuhendislikPro {

    public static int ascii(String satir)
//ascii hesabı yapılıyor

    {

        String          cevir          =
satir.toLowerCase();
```



```

// Dosyadan verileri kucuk harfe cevirerek
al

        int kelime_toplam = 0;

        for (int a = 0; a < cevir.length();
a++)
//Kelimenin  ascii  deęerini  hesaplatma
işlemini yaptık

        {

                char                c_karakter                =
cevir.charAt(a);

                int                ascii_no                =                (int)
c_karakter;

                kelime_toplam +=  ascii_no  *
Math.pow((a + 1), 4);

        }

        return                kelime_toplam;
//ascii                methodunu                int                tipinde

```

kelime_toplam degerini dödürme işlemini
gerçekleştirdik

```
}
```

```
public static String[]  
eksiltili_arama(String satir)  
//Girilen kelime listede yoksa harf harf  
eksiltilerek arama işlemi  
gerçekleştiriliyor
```

```
{
```

```
String kelime = "";  
//kelime adında boş bir string oluşturduk  
  
String dizi[] = new  
String[satir.length()];  
//Kullanıcıdan aldığımız kelimenin boyutu  
kadarlık bir String dizisi açtık
```

```
int sayac = satir.length() - 1;  
//sayac adında bi int dosyası  
açtık ,girilen kelimenin 1 eksiğini atadık
```

```

        while (sayac >= 0)
//sayacın 0dan büyük olduğunda donen br
while dongusu açtık

    {

        for (int i = 0; i <
satir.length(); i++)
//satır uzunluğu büyüklüğünde for dongusu
acıldı

        {

            if (i != sayac) {

                kelime +=
satir.charAt(i);
//Tek tek yeni oluşan kelimeler bulundu

            }

        }

        dizi[sayac] = kelime;
//diziye yeni degerleri atıldı

```

```

        kelime = "";
//kelime degeri boşaltıldı

        sayac--;
//sayac 1 azaltıldı

    }

    return dizi;
//Sonuc olarak yeni oluşturduğumuz dizi
döndürdük

}

```

```

    public static String[]
komsu_arama(String satir)
//Girilen kelime listede yoksa komşuların
yer değiştirilmesiyle oluşan kelimeler
bulundu

```

```

{

```

```

    String kelime = "";

```

```

        String        dizi[]        =        new
String[satir.length() - 1];

        char        karakterDizi[]        =        new
char[satir.length()];

        int sayac = satir.length() - 1;

        char degisken;

        while (sayac > 0) {

                for        (int        k        =        0;        k        <
karakterDizi.length; k++) {

                        karakterDizi[k]                        =
satir.charAt(k);
//diziye satirdkaki harfleri tek tek attık

                }

                degisken                        =
karakterDizi[sayac];
//gecici degiskene harf atadık

```

```

        karakterDizi[sayac] =
karakterDizi[sayac - 1];
//yer değişimi yapıldı

        karakterDizi[sayac - 1] =
degisken;
//gecici kullanıldı

        for (int j = 0; j <
satir.length(); j++)
{
//satır
boyutunda for dongusu acıldı

        kelime += karakterDizi[j];
//yeni kelime oluşturuldu

}

        dizi[sayac - 1] = kelime;
//diziye yeni kelimeler yerleştirildi

        kelime = "";
//kelime boşaltıldı

        sayac--;
//sayac 1 azaltıldı

```

```

        }

        return dizi;
//Oluşan dizi geri döndürüldü

    }

    public static String path =
    "Kelimeler.txt";
//Kelşmelerin bulunduğu dosya global bir
şekilde String degere atılıyor

    public static String[] dosya_okuma()
//Kelimelerin bulunduğu dosya okutuldu

    {

        ArrayList<String> arraylist = new
ArrayList<String>();
//ArrayList oluşturuldu

        try {

```

```
        FileReader      fr      =      new
FileReader(path);
//Yol belirtildi

        BufferedReader  br      =      new
BufferedReader(fr);
//Okuma fonksiyonu çalıştırıldı

        String satir;

        int sayac = 0;

        while          ((satir          =
br.readLine())          !=          null)
//satir null oluncaya kadar donduruldu

        {

                String      cevir          =
satir.toLowerCase();
//kelime nin küçük harflere çevrilmiş hali
atandı

                arraylist.add(cevir);
//arrayliste oluşan kelime eklendi
```



```

        sayac++;

    }

    br.close();

//dosya kapatıldı

    } catch (Exception E)
//Olası ihtimallere karşı dosyanın var
olmama durumuna karşı hata mesajı verşldi

    {

        System.out.println("Dosya
bulanamadi ...");

//Hata mesajı

    }

    String liste[] = new
String[arraylist.size()];
//arraylistin boyutu kadar bir String
dizisi oluşturuldu,geri dönen deger
String[] dizi olduğu için

```

```

        for (int i = 0; i <
arraylist.size(); i++)
{
    //arraylist teki degerlerin
String dizisine ataması gerekleřti

        liste[i] = arraylist.get(i);
//atama

}

return liste;
//String dizisi geri dndrlyot

}

```

```

public static int[] asciiListesi()
//Kelimelerin ascii degerlerine gore mod
alınarak ascii dizisi oluřturuluyor

{

```

```

        int[] secili_dizi = new int[102];
//Ascii degerlerinin modu alınipp
yerleştirilecekleri dizi

        String[] dizi = new
String[dosya_okuma().length];
//dosya_okuma methodundan donen listenin
bouyutu kadar bir dizi döndürülüyor

        dizi = dosya_okuma();
//dosya_okuma dan donen liste yeni
oluşturudumuz diziye aktarılıyor

        int kelime_toplam = 0;
//kelime_toplam ilk degeri 0 atılıyor

        for (int i = 0; i < dizi.length;
i++)
//dizi boyutu kadar donen bir for dongusu

        {

                kelime_toplam =
ascii(dizi[i]); //Bu
for dongüsünde bütün kelimeler tek tek

```

ascii methodunu gönderilerk assci deger toplamları

```
        for (int j = 0; j <
secili_dizi.length; j++)
//bulunarak 202 ye gore modları alınıyor ve
sonrasında oluşturduğumuz diziye
yerleştiriliyorlar
```

```
{ //eger
dizide o alan doluysa quadratic yontemine
gore işlemler gerçekleştirilerek diziye
ekleme yapılmaya
```

```
        int mod = kelime_toplam %
101;
//çalışılıyor.Son olarak oluşan dizi geri
dödürülüyor
```

```
        if (secili_dizi[mod] == 0)
{
        secili_dizi[mod] =
kelime_toplam;
```

```

        break;

    } else {

        int kalan = (mod + (j
* j)) % 101;

        if (secili_dizi[kalan]
== 0) {

secili_dizi[kalan] = kelime_toplam;

            break;

        }

    }

    }

    kelime_toplam = 0;

}

return secili_dizi;

}

```

```

        public static String[] isimListesi()
//ascii listesinde yapılabilirdi fakat
geriye 2 tane dizi döndüremediğimizden
dolayı

{
        //bu methodu da oluşturuyoruz.Bu
methodda da aynı şekilde işlemler
gerçekleştiriliyor.

        String myList[] = new String[102];
//Son olarak bundanda isimlerin listesi
return ediliyor

        String[] dizi = new
String[dosya_okuma().length];

        dizi = dosya_okuma();

        int kelime_toplam = 0;

        for (int i = 0; i < dizi.length;
i++) {

```

```

        kelime_toplam
ascii(dizi[i]);

        for (int j = 0; j <
myList.length; j++) {

            int mod = kelime_toplam %
101;

            if (myList[mod] == null) {

                myList[mod] = dizi[i];

                break;

            } else {

                int kalan = (mod + (j
* j)) % 101;

                if (myList[kalan] ==
null) {

                    myList[kalan] =

dizi[i];

                    break;

```

```

        }

    }

}

    kelime_toplam = 0;

}

return myList;

}

    public static void arama() //Arama
yapacağımız method

    {

        int[] secili_dizi = new int[102];

        String myList[] = new String[102];

```



```

myList = isimListesi();

secili_dizi = asciiListesi();

ArrayList array = new ArrayList();

System.out.println("*****ARAMA
BOLUMU*****");

        System.out.print("Aranacak      adi
giriniz : ");

        Scanner      input      =      new
Scanner(System.in);           //Bu methodda
kullanıcı bir kelime giriyor girilen kelime
listede aratılıyor eger varsa ekrana mesaj
yollanıyor

        String      metin;
//yoksa      kelime      eksiltili_arama      ve
komsu_arama      methodlarına      yollanıyor      ve
donen degerler tekrardan listede

```

```
//varmı diye kontrol
```

```
        metin            =            input.next();  
//ediliyor eger bu iki methoddan gelen  
sonuçlara gore de listede yoklarsa hiç bir  
şekilde kelimenin olmadığı yönünde mesaj  
gönderiliyor.
```

```
        int kelime_toplam = 0;
```

```
        String            cevir            =  
metin.toLowerCase();
```

```
        String            dizi[]            =            new  
String[cevir.length()];
```

```
        for (int i = 0; i < myList.length;  
i++) {
```

```
                array.add(myList[i]);
```

```
        }
```

```
        kelime_toplam = ascii(cevir);
```

```
String[] eksiltili_dizi = new
String[cevir.length()];
```

```
String[] komsu_dizi = new
String[cevir.length() - 1];
```

```
boolean deger = false;
```

```
for (int i = 0; i <
secili_dizi.length; i++) {
```

```
    if (secili_dizi[i] ==
kelime_toplam)
```

```
// kelimenin ascci degeriyle girilen
kelimenin ascci degeri karşılaştırılıyor
```

```
{
```

```
// eger uygunsa listede bu kelime aynı
isimle de bulunuyormu diye kontrol ediliyor
```

```
System.out.println("
Girdiginiz *" + metin + "*" kelimesi metin
```

```

dosyasinda                bulunmaktadir...");
//kelime  metin  dosyasinda  her  ihtimale
karşı var

                deger = true;

        }

}

int sayac = 0;

if (deger == false) {

        for    (int    i    =    0;    i    <
secili_dizi.length; i++) {

                eksiltili_dizi                =
eksiltili_arama(cevir);

                komsu_dizi                =
komsu_arama(cevir);

                for    (int    j    =    0;    j    <
eksiltili_dizi.length; j++) {

```

```

            if
(ascii(eksiltili_dizi[j]) ==
secili_dizi[i]) {

System.out.println("Girdiğiniz *" + metin
+ "*" kelimesi metin dosyasında *" +
eksiltili_dizi[j] + "*" olarak
bulunmuştur");

            sayac++;

//Sayac kullanmamızın sebebi kelimeni
listede hiç bulunamama durumunda kontrol
sayacı olarak görev almıştır

            break;

        }

    }

    for (int j = 0; j <
komsu_dizi.length; j++) {

```

```

        if
(ascii(komsu_dizi[j]) == secili_dizi[i]) {

System.out.println("Girdiğiniz *" + metin
+  "*" kelimesi metin dosyasında "*" +
komsu_dizi[j] + "*" olarak bulunmuştur");

        sayac++;

//yukarıdaki sayac ile aynı.Bu sayac eger
0'da kalmassa demekki hiç bu if lere
girmemiştir demektir

        break;

//İşlemi sonlandırma kısmı sayaca gerek
kalmıyor gibi gözüksede olmadığında hata
veriyor

    }

}

```

```

    }

    if (sayac == 0) {

        System.out.println("*" +
        metin + "*" ile alakalı metin dosyasında
        benzer kelime bulunamamıştır ...");
        //sayacın 0 ve kelimenin listede hiç olmama
        durumunda çalışıyor

    }

    sayac = 0;

}

}

//Yerleştirdiğimiz kelimelerin
indisi,kelime adı,ascii karşılıklarını
ekrana yazdırma işlemi

public static void listeYaz() {

    String myList[] = new String[102];

```

```

        myList = isimListesi();

        for (int i = 0; i < myList.length;
i++) {

            if (myList[i] != null) {

                System.out.println "[" + i
+ "]" + ":" + myList[i] + " " +
ascii(myList[i]));

            }

        }

    }

```

```

    public static void main(String[] args)
    {

        listeYaz();
        //listeyaz() methodu çağırılıyor
    }

```



```

        boolean    deger    =    true;
//true , false degerlerini alan bir boolean
tipinde degişken oluşturuluyor

        Scanner    oku    =    new
Scanner(System.in);
//Scanner kutuphanesi

        try    //Klavyeden true , false
dışında deger girme durumu tespit etme

        {

                while (deger == true) //deger
== true olduğu sürece tekrar etsin

                {

                        arama();
//arama() methodu çağırılıyor

                        System.out.println("Arama
yapmak İstiyormusunuz : (True/False)");
//Kullanıcıdan true ve ya false degeri
girmesi isteniyor

```

```
        deger = oku.nextBoolean();  
//deger kullanıcından gelen mesaj atılıyor  
  
    }  
  
    } catch (Exception E) {  
  
        System.out.println("İstek  
Dışında          deger          girildi");  
//Hata mesajı veriliyor  
  
    }  
  
    }  
  
}
```