

```
////////////////////////////////////  
//////////////////////////////////// ALGORİTMA VE PROGRAMLAMA DERSİ-2 /// ODEV-1 //////////////////////////////////////  
////////////////////////////////////
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <ctype.h>  
void buyukHarfe_cevir(char kelime[])//karsilasitmada kolaylik saglamak amaciyla  
{  
    //yapilmistir, harfleri buyuk harfe ceviren fonksiyon  
    int i=0;  
  
    for(i=0; i<strlen(kelime); i++)//kelime boyutu kadar donme saglaniyor..  
    { switch(kelime[i])  
      {  
        case (char)135: kelime[i]=(char)128; break;//Turkce karakterlerin buyuk harfe cevrilmesi...  
        case (char)167: kelime[i]=(char)166; break;  
        case (char)141: kelime[i]=(char)73; break;  
        case (char)105: kelime[i]=(char)152; break;  
        case (char)148: kelime[i]=(char)153; break;  
        case (char)159: kelime[i]=(char)158; break;  
        case (char)129: kelime[i]=(char)154; break;  
        default : kelime[i]=toupper(kelime[i]); break;//kalan karakterler buyuk harfe cevriliyor...  
      }  
    }  
}  
  
int karakter_ara(char harf)//kullanıcının girdiği karakterlerin Turkce olup olmadigi kontrol ediliyor..  
{  
    char *konum3;  
    char *alfabe="ABCCDEFGĞHİİJKLMNOÖPRSŞTUÜVYZ";  
    switch(harf)  
    { case (char)128 : konum3=strchr(alfabe,'Ç'); break;//Turkce karakterler stringte araniyor...  
      case (char)166 : konum3=strchr(alfabe,'Ğ'); break;  
      case (char)152 : konum3=strchr(alfabe,'İ'); break;  
      case (char)153 : konum3=strchr(alfabe,'Ö'); break;  
      case (char)158 : konum3=strchr(alfabe,'Ş'); break;  
      case (char)154 : konum3=strchr(alfabe,'Ü'); break;  
      default      : konum3=strchr(alfabe,harf);break;//ardından diğerkarakterler stringte aranıyor..  
    }  
    if(konum3=='\0')//Eger karakter stringte yer almıyorsa degisken donduruluyor..  
        return 1; //Bu degisken kullanıcının Turkce karakter girip girmediğini belirtiyor..  
    else  
        return 0;  
}  
  
int AlfabetikSiral(char Turkce_kelime[],char yeniKelime[])//iki kelimedenden hangisinin alfabetik olarak  
once  
{ //geldiğini bulmak amacıyla yazıldı..  
    int i;  
    char *alfabe="ABCCDEFGĞHİİJKLMNOÖPRSŞTUÜVYZ";//Turkce alfabede sıra belirlemek için
```

```

char
*sayisalalfabe="1011121314151617181920212223242526272829303132333435363738";//sayisal
bir alfabe de kullanildi...
char *konum;
char *konum2;
char *sayisalkonum;
char *sayisalkonum2;
char sayisalkelime[5];
char sayisalkelime2[5];
int min;
int sir;
if(strlen(Turkce_kelime)>strlen(yeniKelime))//kelimelerin uzunluklari belilenererek
    min=strlen(yeniKelime);          //kisa olanin sonuna kadr karsilastirma yapildi..
else                                //fazladan karakter okunmamasini icin...
    min=strlen(Turkce_kelime);

for(i=0; i<min; i++)//kelimelerden herhangi biri NULL olmadigi surece
{ switch(Turkce_kelime[i])
  { case (char)128 : konum=strchr(alfabe,'Ç'); break;//Turkce karakterlerin
    case (char)166 : konum=strchr(alfabe,'Ğ'); break; //konumu belirleniyor...
    case (char)152 : konum=strchr(alfabe,'İ'); break;
    case (char)153 : konum=strchr(alfabe,'Ö'); break;
    case (char)158 : konum=strchr(alfabe,'Ş'); break;
    case (char)154 : konum=strchr(alfabe,'Ü'); break;
    default      : konum=strchr(alfabe,Turkce_kelime[i]);break;//kalan karakterlerin konumu
belirleniyor...
  }
  switch(yeniKelime[i])
  { case (char)128 : konum2=strchr(alfabe,'Ç'); break;
    case (char)166 : konum2=strchr(alfabe,'Ğ'); break;
    case (char)152 : konum2=strchr(alfabe,'İ'); break;
    case (char)153 : konum2=strchr(alfabe,'Ö'); break;
    case (char)158 : konum2=strchr(alfabe,'Ş'); break;
    case (char)154 : konum2=strchr(alfabe,'Ü'); break;
    default      : konum2=strchr(alfabe,yeniKelime[i]);break;
  }

  sayisalkonum=(konum-alfabe)*2+sayisalalfabe;//sayisalalfabeden yararlanilarak
  sayisalkonum2=(konum2-alfabe)*2+sayisalalfabe;//degerleri daha kolay karsilastirilabilcek
stringler elde ediliyor...
strncpy(sayisalkelime,sayisalkonum,2);
sayisalkelime[2]='\0';//her harf icin 2 basamakli sayilar var...
strncpy(sayisalkelime2,sayisalkonum2,2);
sayisalkelime2[2]='\0';          //VE KARSILASTIRMALAR...
if(strcmp(sayisalkelime,sayisalkelime2)<0)//alfabetik olarak onde olan kelimeler yaziliyor..
{
    sir=1;//turkce kelime yeni kelimedenden ondeki bir sirada
    break;
    break;
}
else if(strcmp(sayisalkelime,sayisalkelime2)>0 )
{

```

```

        sira=-1;//turkce kelime yeni kelimedden gerideki bir sirada
        break;
        break;
    }
    else if(strcmp(yeniKelime,Turkce_kelime)==0 )
    {
        sira=0;//ikisi ayni kelime
    }
}
return sira;//elde edilen deger donduruluyor...
}
void kelime_ekle()//dosyaya kelime ekleniyor...
{
    char Turkce_kelime[21];
    char yeniKelime[21];
    char Ingilizce_anlami[101];
    char yeniAnlam[101];
    char bosluk;
    int hata;//karakter_ara fonksiyonunun dondurdugu degeri tutan degisken
    int yazildi;// yazilan satirin karsilastirma sirasinda tekrar yazilmasini engellemek amaciyla...
    int j,ayni;
    int i,sira;
    FILE *dosya;
        FILE *dosya2;

if((dosya = fopen("sozluk.dat","r")) == NULL)//dosyanin acilmama durumu...
    printf("sozluk dosyasi acilamadi!\n");
else{
        //acilrsa
        printf("Ekleme istediginiz kelimeyi girin:\n");

        for(j=0; j<2;j++)//kelime aliniyor ve kontrol ediliyor...
        { gets(yeniKelime);
            buyukHarfe_cevir(yeniKelime);//buyuk harfe ceviriliyor...
            for(i=0;i<strlen(yeniKelime);i++)
            { hata=karakter_ara(yeniKelime[i]);//kontrol...
                if(hata==1)//hatali giris icin
                {
                    printf("Turkce karakter kullanin ve bosluk kullanmayin.\n"
                        "Tekrar giris yapin:\n" );
                    j-=1;//tekrar kelime istenmesi icin for dongusu degiskeni bir azaltiliyor...
                    break;
                }
            }
        }
    }

    fscanf(dosya,"%s",Turkce_kelime);
    bosluk=fgetc(dosya);
    fgets(Ingilizce_anlami,101,dosya);
    buyukHarfe_cevir(Turkce_kelime);
    sira=AlfabetikSiralama(Turkce_kelime,yeniKelime);//dosyadan alinan kelime ile kullanicinin
    ayni=0; //ayni olma durumunda uyarı yapabilmek için... //girdigi karsilastiriliyor...

```

```

while(!feof(dosya))
{
    if(sira==0)//ayni kelimeler
        ayni=1;
    else if(sira==-1)
        break;//arama biter

    fscanf(dosya,"%s",Turkce_kelime);
    bosluk=fgetc(dosya);
    fgets(Ingilizce_anlami,101,dosya);
    buyukHarfe_cevir(Turkce_kelime);
    sira=AlfabetikSiralama(Turkce_kelime,yeniKelime);
}
fclose(dosya);//tum dosyadaki kelimelerle karsilastirdi
}
if(ayni==1)//donguden cikildiginda ayni 1 ise uyarı verilir
    printf("Girdiginiz kelime bulunmaktadir.\n");
else
    if(ayni==0)// siranin 0 a esit olmadigi durum (kelimeler ayni degilse)...
    { dosya = fopen("sozluk.dat","r");//ayni kelimeye rastlanmadiysa gecici dosya da acilarak ekleme yapiliyor...
        dosya2 = fopen("gecici.dat","w");
        fscanf(dosya,"%s",Turkce_kelime);//dosyadan veri alinisi...
        bosluk=fgetc(dosya);
        fgets(Ingilizce_anlami,101,dosya);
        buyukHarfe_cevir(Turkce_kelime);

        printf("Kelimenin anlamini girin.\n");
        gets(yeniAnlam);
        sira=AlfabetikSiralama(Turkce_kelime,yeniKelime);//eklenecek kelimenin sirasi belirleniyor..
        printf("Yeni kelime eklenmistir.\n");
        yazildi=0;//henuz yeni kelime dosyaya yazilmadi..
        while(!feof(dosya))
        {
            if(sira==-1 && yazildi==1)// kelime eklendikten sonrakiler dosyaya yaziliyor...
            {
                fprintf(dosya2,"%s %s",Turkce_kelime,Ingilizce_anlami);
            }
            if(sira==1)//eklenecek kelimedenden ondekiler yaziliyor...
                fprintf(dosya2,"%s %s",Turkce_kelime,Ingilizce_anlami);
            if(sira==-1 && yazildi==0)//yeni kelime ekleniyor
            { fprintf(dosya2,"%s %s\n",yeniKelime,yeniAnlam);
                fprintf(dosya2,"%s %s",Turkce_kelime,Ingilizce_anlami);
                yazildi=1;// tekrar yazimi engellemek amacıyla degiskene 1 atanir...
            }
        }

        fscanf(dosya,"%s",Turkce_kelime);//dosya sonuna kadr islem devam eder...
        bosluk=fgetc(dosya);
        fgets(Ingilizce_anlami,101,dosya);
        buyukHarfe_cevir(Turkce_kelime);
        sira=AlfabetikSiralama(Turkce_kelime,yeniKelime);
    }
}

```

```

        fclose(dosya);//dosyalar kapatilir..
        fclose(dosya2);
        remove("sozluk.dat");//dosya siliniyor...
        rename("gecici.dat","sozluk.dat");//gecici dosyanin adi degistiriliyor...
    }
}
void tumKelimleri_listele();//dosyadaki tum kelimeler listelenir...
{ char Turkce_kelime[21];
  char Ingilizce_anlami[101];
  char bosluk,c;
  FILE *dosya;
  int i,j;

  if((dosya = fopen("sozluk.dat","r")) == NULL)//dosyanin acilmama durumu...
    printf("sozluk dosyasi acilamadi!\n");
  else{
      //acilrsa...
      fscanf(dosya,"%s",Turkce_kelime);
      bosluk=fgetc(dosya);
      fgets(Ingilizce_anlami,101,dosya);

      while(!feof(dosya))//alınan veriler dosya sonu gelinceye kadar sirayla yazdiriliyor...
      {
          for(i=0; i<2; i++)
          {
              printf("TURKCE KELİME    INGILIZCE ANLAMI\n");
              printf("===== \n");

              for(j=0; j<(i+1)*16; j++)//sayfalama islemi yapiliyor...
              {
                  printf("%-19s%s",Turkce_kelime,Ingilizce_anlami);
                  fscanf(dosya,"%s",Turkce_kelime);
                  bosluk=fgetc(dosya);
                  fgets(Ingilizce_anlami,101,dosya);
                  if(feof(dosya))
                  { break;
                    }
                }
              if(i!=1)
              { printf("\nBir sonraki sayfaya gecmek icin herhangi bir tusa basin.\n\n");
                fflush(stdin);
                c=getchar();
              }
            }
          }
        }
      }
    fclose(dosya);//isi biten dosya kapatilir...
}
void anlam_guncelle();//var olan bir kelimenin anlami degistiriliyor...
{
  char Turkce_kelime[21];
  char Ingilizce_anlami[101];
  char yeniKelime[21];

```

```

char yeniAnlam[101];
char bosluk;
int hata;
int i,j;
int sirasira;
int ayni=0;//ayni kelime yok
FILE *dosya;
FILE *dosya2;

if((dosya = fopen("sozluk.dat","r")) == NULL)//dosyanin acilmama durumu...
    printf("sozluk dosyasi acilamadi!\n");
else{
    if((dosya2 = fopen("gecici.dat","w"))== NULL)//gecici dosyanin acilmama durumu...
        printf("Gecici dosya acilamadi!\n");
    else//ikisi de acilirsasira...
    {
        printf("Guncellemek istediginiz kelimeyi girin.\n");

        for(j=0; j<2;j++)//girilen kelimenin karakterleri kontrol ediliyor...
        { gets(yeniKelime);
          buyukHarfe_cevir(yeniKelime);
          for(i=0;i<strlen(yeniKelime);i++)
          { hata=karakter_ara(yeniKelime[i]);//hatali olup olmadigi donduruluyor...
            if(hata==1)
            {
                printf("Turkce karakter kullanin ve bosluk kullanmayin.\n"
                    "Tekrar giris yapin:\n" );
                j-=1;//tekrar giris yapilmasi saglaniyor...
                break;
            }
          }
        }
        fscanf(dosya,"%s",Turkce_kelime);//dosyadan veriler aliniyor...
        bosluk=fgetc(dosya);
        fgets(Ingilizce_anlami,101,dosya);
        buyukHarfe_cevir(Turkce_kelime);
        sirasira=AlfabetikSirala(Turkce_kelime,yeniKelime);//alfabetik oncelik belirleniyor...
        while(!feof(dosya))
        { if(sira==1)//girilen kelimedenden once olanlar yazdiriliyor gecici dosyaya...
          fprintf(dosya2,"%s %s",Turkce_kelime,Ingilizce_anlami);
          else if(sira==0)//kelime bulununca yeni anlam yazdiriliyor...
          { printf("Yazdiginiz kelime bulunmaktadir.\n");
            printf("Bulunan anlam: %s\n",Ingilizce_anlami);
            printf("Yeni anlami girin:\n");
            gets(yeniAnlam);
            fprintf(dosya2,"%s %s\n",Turkce_kelime,yeniAnlam);
            ayni=1;//ayni kelimenin olup olmadigi degiskene ataniyor..
          }
          else//sonrakiler yazdiriliyorr...
          {
              fprintf(dosya2,"%s %s",Turkce_kelime,Ingilizce_anlami);
          }
        }
    }
}

```

```

        fscanf(dosya,"%s",Turkce_kelime);
        bosluk=fgetc(dosya);
        fgets(Ingilizce_anlami,101,dosya);
        buyukHarfe_cevir(Turkce_kelime);
        sirasira=AlfabetikSiralama(Turkce_kelime,yeniKelime);
    }//yazma isi biter ve dosyalar kapatilir...
    fclose(dosya);
    fclose(dosya2);
    remove("sozluk.dat");//dosya siliniyor...
    rename("gecici.dat","sozluk.dat");//gecici dosyanin adi degistiriliyor...
    if(ayni==1)//kullanici bilgilendiriliyor...
        printf("Girdiginiz kelimenin anlami guncellenmistir.\n");
    else
        printf("Girdiginiz kelime bulunmadigindan anlam guncellenemedi\n");
}

}

}
void anlam_sorgula()//girilen kelimenin anlami sorgulanir...
{
    char Turkce_kelime[21];
    char Ingilizce_anlami[101];
    char yeniKelime[21];
    char bosluk;
    int hata,i,j;
    int sirasira;
    int ayni=0;//ayni kelime yok
    FILE *dosya;

    if((dosya = fopen("sozluk.dat","r")) == NULL)//dosyanin acilmama durumu...
        printf("sozluk dosyasi acilamadi!\n");
    else{
        //acilirsasira...
        printf("Anlamini sorgulamak istediginiz kelimeyi girin.\n");

        for(j=0; j<2;j++)//girilen kelimenin karakterleri kontrol ediliyor...
        {
            gets(yeniKelime);
            buyukHarfe_cevir(yeniKelime);
            for(i=0;i<strlen(yeniKelime);i++)
            {
                hata=karakter_ara(yeniKelime[i]);
                if(hata==1)
                {
                    printf("Turkce karakter kullanin ve bosluk kullanmayin.\n"
                        "Tekrar giris yapin:\n" );
                    j=1;//hata varsa tekrar giris yaptiriliyor...
                    break;
                }
            }
        }
    }

    fscanf(dosya,"%s",Turkce_kelime);//dosyadan kelimeler aliniyor...
    bosluk=fgetc(dosya);
    fgets(Ingilizce_anlami,101,dosya);

```

```

buyukHarfe_cevir(Turkce_kelime);
sira=AlfabetikSiralama(Turkce_kelime,yeniKelime);//karsilastirma islemi...
while(!feof(dosya))
{
    if(sira==1)//girilen harfe gore alfabetik olarak kelimeler gerideyse
    {
        //arama biter...
        break;
    }
    if(sira==0)//iki kelimenin ayni olma durumu....
    {
        printf("Girdiginiz kelimenin ingilizce anlami:\n"
            "%s",Ingilizce_anlami);
        ayni=1;
        break;//ayni kelime bulunuca arama biter...
    }
    fscanf(dosya,"%s",Turkce_kelime);
    bosluk=fgetc(dosya);
    fgets(Ingilizce_anlami,101,dosya);
    buyukHarfe_cevir(Turkce_kelime);
    sira=AlfabetikSiralama(Turkce_kelime,yeniKelime);
}
if(ayni==0)//ayni kelime yoksa kullanıcı bilgilendiriliyor...
    printf("Girdiginiz kelime dosyada bulunmadigindan anlami sorgulanamamistir.\n");
fclose(dosya);//islem biter dosya kapatilir...
}
}

void harfeGore_listele();//girilen bir harfe gore e listeleme islemi...
{
    //ilk harf ele alinir...
    char harf[1];
    char Turkce_kelime[21];
    char Ingilizce_anlami[101];
    int hata,i,j;
    int sira;
    int ayni=0;//ayni kelime yok
    FILE *dosya;
    char bosluk;

    if((dosya = fopen("sozluk.dat","r")) == NULL)//dosyanin acimama durumu...
        printf("sozluk dosyasi acilamadi!\n");
    else{
        printf("Harfi girin.\n");

        for(j=0; j<2;j++)//kelimenin kontrolu...
        { gets(harf);
            buyukHarfe_cevir(harf);
            for(i=0;i<strlen(harf);i++)
            { hata=karakter_ara(harf[i]);
                if(hata==1)
                {
                    printf("Turkce karakter kullanin ve bosluk kullanmayin.\n"
                        "Tekrar giris yapin:\n" );
                    j-=1;
                }
            }
        }
    }
}

```



```

        break;
    }
}
}
printf("TURKCE KELİME    İNGİLİZCE ANLAMI\n");
printf("===== \n");

fscanf(dosya,"%s",Turkce_kelime);//dosyadan veri alimi...
bosluk=fgetc(dosya);
fgets(Ingilizce_anlami,101,dosya);
buyukHarfe_cevir(Turkce_kelime);
sira=AlfabetikSirala(Turkce_kelime,harf);//alfabetik olarak karsilastirma islemi...
while(!feof(dosya))
{
    if(sira==-1)//girilen harfe gore alfabetik olarak kelimeler gerideyse
    {
        //arama biter...
        break;
    }
    if(Turkce_kelime[0]==harf[0])//ilk harfin ayni olma durumu...
    {
        //VE LİSTELEME...
        printf("%-19s %s",Turkce_kelime,Ingilizce_anlami);
        ayni=1;
    }
    fscanf(dosya,"%s",Turkce_kelime);
    bosluk=fgetc(dosya);
    fgets(Ingilizce_anlami,101,dosya);
    buyukHarfe_cevir(Turkce_kelime);
    sira=AlfabetikSirala(Turkce_kelime,harf);
}
if(ayni==0)//Eger karsilastirmalarda bulunamadiysa kullanıcı bilgilendirilir...
    printf("Girdiginiz harf ile baslayan kelime bulunmadigindan anlami sorgulanamamistir.\n");

fclose(dosya);//islen biter dosya kaatilir...
}
}
void metneGore_listele()//girilen bir metin parcasini iceren stringler listelenir...
{
    char Turkce_kelime[21];
    char Ingilizce_anlami[101];
    char yeniKelime[21];
    char bosluk;
    int hata,i,j;
    int ayni=0;//ayni kelime yok
    FILE *dosya;

    if((dosya = fopen("sozluk.dat","r")) == NULL)//dosyanin acilmama durumu...
        printf("sozluk dosyasi acilamadi!\n");
    else{
        printf("Metni girin.\n");

        for(j=0; j<2;j++)//metin karakter kontrolu...
        { gets(yeniKelime);

```

```

        buyukHarfe_cevir(yeniKelime);
        for(i=0;i<strlen(yeniKelime);i++)
        { hata=karakter_ara(yeniKelime[i]);
          if(hata==1)
          {
              printf("Turkce karakter kullanin ve bosluk kullanmayin.\n"
                    "Tekrar giris yapin:\n" );
              j-=1;
              break;
          }
        }
    }
    printf("TURKCE KELİME      INGILIZCE ANLAMI\n");
    printf("=====      =====\n");

    fscanf(dosya,"%s",Turkce_kelime);
    bosluk=fgetc(dosya);
    fgets(Ingilizce_anlami,101,dosya);
    buyukHarfe_cevir(Turkce_kelime);//buyuk harfe cevirme...
    while(!feof(dosya))
    {
        if(strstr(Turkce_kelime,yeniKelime)!=NULL)//string icinde string araniyor...
        {
            //ve listeleme islemi...
            printf("%-19s %s",Turkce_kelime,Ingilizce_anlami);
            ayni=1;
        }
        fscanf(dosya,"%s",Turkce_kelime);
        bosluk=fgetc(dosya);
        fgets(Ingilizce_anlami,101,dosya);
        buyukHarfe_cevir(Turkce_kelime);
    }
    if(ayni==0)//metnin bulunamamasi durumunda kullanıcı bilgilendiriliyor...
        printf("Girdiginiz metin parcasinin gectigi kelime olmadigindan listeleme yapilmamistir.");

    fclose(dosya);//islem biter dosya kapatilir...
}
}
void istatistik_listele();//hangi harfin hangi sirada kac kez ciktiginin listelenmesi...
{
    int i;
    char *alfabe="ABCÇDEFGĞHIİJKLMNOÖPRSŞTUÜVYZ";
    char *konum;
    char Turkce_kelime[21];
    char Ingilizce_anlami[101];
    char bosluk,c;
    int a,b;
    int toplam[29]={0};//her harf toplami icin...
    FILE *dosya;
    int istatistik[29][20]={0};//harf ve konum belirtildigi cift boyutlu dizi kullaniliyor...
    if((dosya = fopen("sozluk.dat","r")) == NULL)//dosyanin acilmama durumu...
        printf("sozluk dosyasi acilamadi!\n");
    else{

```

```

        //acilirsa...
fscanf(dosya,"%s",Turkce_kelime);
bosluk=fgetc(dosya);
fgets(Ingilizce_anlami,101,dosya);
buyukHarfe_cevir(Turkce_kelime);
while(!feof(dosya))
{
    for(i=0; i<strlen(Turkce_kelime); i++)
    {
        switch(Turkce_kelime[i])
        {
            case (char)128 : konum=strchr(alfabe,'Ç'); break;//Turkce karakterlerin alfabedeki
            case (char)166 : konum=strchr(alfabe,'Ğ'); break;//siralardan yararlaniliyor...
            case (char)152 : konum=strchr(alfabe,'İ'); break;//Bu islem ile harfler belirleniyor...
            case (char)153 : konum=strchr(alfabe,'Ö'); break;
            case (char)158 : konum=strchr(alfabe,'Ş'); break;
            case (char)154 : konum=strchr(alfabe,'Ü'); break;
            default : konum=strchr(alfabe,Turkce_kelime[i]); break;
        }

        a=(int)konum-(int)alfabe;//kacinci siradaki harf oldugu bulunuyor...
        b=i;//kelimedeki konum...
        istatistik[a][b]++;//her seferinde bir artirilerek sayim yapiliyor...
    }
    fscanf(dosya,"%s",Turkce_kelime);
    bosluk=fgetc(dosya);
    fgets(Ingilizce_anlami,101,dosya);
    buyukHarfe_cevir(Turkce_kelime);
}
printf("Harf 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 Toplam\n");
printf("-----\n");
i=0;
for(a=0; a<29; a++)
{
    switch(a){
        case 3: printf(" %c ",(char)128); break;//Turkce karakterler yazdiriliyor...
        case 8: printf(" %c ",(char)166); break;
        case 11:printf(" %c ",(char)152); break;
        case 18:printf(" %c ",(char)153); break;
        case 22:printf(" %c ",(char)158); break;
        case 25:printf(" %c ",(char)154); break;
        default:printf(" %c ",alfabe[a]); break;//digerleri yazdirilir....
    }
    for(b=0; b<20; b++)//harf toplmi bulunuyor ve konuma gore yazdirma islemi yapiliyor...
    {
        printf("%-3d",istatistik[a][b]);
        toplam[a]+=istatistik[a][b];
    }
    printf("%3d\n",toplam[a]);
    i++;
    if(i==14)//sayfalama...
    {
        printf("\nBir sonraki sayfaya gecmek icin herhangi bir tusa basin.\n\n");
        fflush(stdin);
    }
}

```

```

        c=getchar();
        printf("Harf 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 Toplam\n");
        printf("-----\n");
    }
}

fclose(dosya);//islem biter dosya kapatilir....
}
}
int menu_secim(void)//menu yazdiriliyor....
{
    int secim;

    printf("\n      ===== MENU =====\n\n");
    printf("    1. Yeni Kelime Ekleme\n\n");
    printf("    2. Sozlukteki Bir Kelimenin Ingilizce Anlamini Guncelleme\n\n");
    printf("    3. Sozlukteki Bir Kelimenin Ingilizce Anlamini Sorgulama\n\n");
    printf("    4. Girilen bir Turkce harf ile baslayan kelimelerin ve\n"
        "        Ingilizce anlamlarinin listelenmesi\n\n");
    printf("    5. Girilen bir Turkce metin parcasinin gectigi kelimelerin \n"
        "        ve Ingilizce anlamlarinin listelenmesi\n\n");
    printf("    6. Sozlukteki tum kelimelerin ve Ingilizce anlamlarinin\n"
        "        listelenmesi\n\n");
    printf("    7. Harflerin kelime icinde kullanildiklari yer\n"
        "        istatistiklerinin listelenmesi\n\n");
    printf("    8. Cikis \n\n");
    printf("      =====\n\n");
    do {
        printf("\nSeciminizi Giriniz:\n");//secim dogru girilinceye kadar bekleniyor...
        scanf("%d", &secim);
    } while(secim<1 || secim>8);

    return secim;//secim gonderiliyor...
}
int main(void)
{
    int secenek;//secime gore uygun fonk cagiriliyor....
    char cikis,c;
    do{
        cikis='h';
        secenek=menu_secim();
        switch(secenek) {
            case 1:
                kelime_ekle();//yeni kelime ekler...
                printf("\nMenuye donmek icin herhangi bir tusa basin.\n");
                fflush(stdin);
                c=getchar();
                break;
            case 2:
                anlam_guncelle();//var olan kelimenin anlamini yeniler...
                printf("\nMenuye donmek icin herhangi bir tusa basin.\n");
                fflush(stdin);

```

```

        c=getchar();
        break;
    case 3:
        anlam_sorgula();// kelimenin anlamini yazdirir...
        printf("\nMenuye donmek icin herhangi bir tusa basin.\n");
        fflush(stdin);
        c=getchar();
        break;
    case 4:
        harfeGore_listele();//girilen harfi ilk harfinde barindiran kelimeler yazdirilir...
        printf("\nMenuye donmek icin herhangi bir tusa basin.\n");
        fflush(stdin);
        c=getchar();
        break;
    case 5:
        metneGore_listele();//girilen metni iceren kelimeler yazdirilir...
        printf("\nMenuye donmek icin herhangi bir tusa basin.\n");
        fflush(stdin);
        c=getchar();
        break;
    case 6:
        tumKelimleri_listele();//sirayla dostadaki tum kelimeler listelenir...
        printf("\nMenuye donmek icin herhangi bir tusa basin.\n");
        fflush(stdin);
        c=getchar();
        break;
    case 7:
        istatistik_listele();//Turkce kelimedeki harf istatistigi...
        printf("\n\nMenuye donmek icin herhangi bir tusa basin.\n\n");
        fflush(stdin);
        c=getchar();
        break;
    case 8:
        do{//cevap dogru girilinceye kadar bekleniyor...
            printf("Cikmak istediginize emin misiniz(e/E/h/H)?:\n");
            fflush(stdin);
            cikis=getchar();
        }while (cikis != 'e' && cikis != 'E' && cikis!='h' && cikis!='H');
        break;
    }
}while(cikis == 'H' || cikis == 'h');

return 0;
}

```