

# DokuWiki Uygulaması

1<sup>st</sup> Tuna Gülcan

Bilgisayar Mühendisliği

Programlama Laboratuvarı-I 1.Proje

İzmit/KOCAELİ

tuna.gulcan@outlook.com

2<sup>nd</sup> Fahrettin Doğan

Bilgisayar Mühendisliği

Programlama Laboratuvarı-I 1.Proje

İzmit/KOCAELİ

fahrettinndogann7@gmail.com

Bu rapor DokuWiki uygulamasıyla alakalı yalancı kod, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmaktadır.

## GİRİŞ

Bu rapor **DokuWiki** uygulamasının kullanımı ve ne işe yaradığıyla ilgili bilgilendirme raporudur.

## YÖNTEM

Uygulamamız bir menüye sahiptir ve kullanıcılarımızın uygulamayı açtığı an itibariyle adım adım yapmak istediği işlemle ilgili her türlü bilgiye ulaşabileceği bir menüdür. Menüümüz 4 farklı seçenek sunmaktadır. Bu seçenekler şu şekildedir. 1.Kelime veya Etiket Arama 2.Güncelleme Yapma 3. Uygulamanın çalıştığı konsolu temizleme. 4. seçenek ise uygulamadan çıkmaktır.

## UYGULAMANIN KULLANIMI

Öncelikle uygulamamız ilk çalıştırıldığında 4 adet seçenek sunulmaktadır. Bu seçeneklerden birini seçmeniz gerekmektedir. Seçiminize göre farklı bir adıma yönlendirileceksiniz. Ve bu adımları nasıl kullanacağınızı ekranda göreceksiniz. Program sizi yönlendirecek. Daha ayrıntılı olarak bu bölümün devamında nasıl kullanıldığının açıklaması bulunmaktadır.

### *Kelime veya Etiket Araması Yapma*

Birinci seçeneğimiz kelime veya etiket araması yapma. Kullanıcı bu seçeneği seçtiği zaman kullanıcıdan bir kelime girdisi isteniyor. Ve kullanıcı bu kelimeyi girdiği zaman uygulama dosyaların içinde tek tek dolaşarak girilen kelimenin bir etiket mi yoksa sadece kelime mi olduğunu buluyor. Ardından ekrana aranan kullanıcı girdisi etiket ise o girdinin etiket olduğunu söylüyor. Kullanıcıdan alınan girdi etiket değil normal bir kelime ise girdinin kelime olduğunu söylüyor. Eğer hiçbir dosya içerisinde bulunamazsa kullanıcıdan alınan girdinin dosyalar içerisinde geçmediğini ekrana bastırıyor. Buna ek olarak girilen etiket veya kelimenin hangi dosyaların içerisinde hangi satırlarda geçtiğini kullanıcıya gösteriyor.

### *Güncelleme*

İkinci seçeneğimiz güncelleme yapma. Bu seçeneğimiz kendi içinde üçe ayrılıyor. Güncellemenin ilk seçeneği olan etiket ve değiştirilen etiketin .txt uzantılı dosya adını değiştirme. İkinci seçeneğimiz yetim etiketler için .txt uzantılı dosya oluşturma. Ve bu dosyaları oluştururken dersin adını ve dersin kodunu dosyanın içine şablon şeklinde oluşturuyor. Son seçeneğimiz olan üçüncü seçenek ise elimizdeki .txt uzantılı dosyaların isimlerini değiştirme.

### *Konsolu Temizleme*

Üçüncü seçeneğimiz konsolu temizleme konsol üzerinde çıkış seçeneğini kullanana kadar uygulamamız sürekli çalışacağı için konsolda bir sürü bilgi olacak bunu önlemek ve kullanıcıya daha rahat bir kullanım sunmak için bu seçeneği seçerek daha önceden yaptığı işlemleri ekrandan temizleyerek. Tekrardan yapmak istediği işlemi yapabilir.

### *Çıkış*

Menünün son seçeneği çıkış yapmaktır. Bu seçeneğin olma nedeni menümüz bir döngü şeklinde çalışmaktadır ve kullanıcı çıkış seçeneğini seçene kadar sürekli olarak yapılan her işlemten sonra kullanıcının tekrardan bir işlem yapması için menüyü ekrana bastırıyor. Eğer kullanıcı başka bir işlem yapmak istemiyorsa bu seçenek sayesinde uygulamadan çıkış yapabiliyor.

## OUTPUT

### *İlk OUTPUT*

**OUTPUT** kısmı uygulamanın console ekranında gözükmemekte. Fakat dosyaların içerisinde bulunun normal ve yetim etiketlerin hepsini bir .txt uzantılı dosyaya atıyor. Diğer bir özelliği ise her bir etiketin dosyalar içerisinde kaç defa geçtiğini gösteriyor.

### *Güncel OUTPUT*

Güncel **OUTPUT** başlığı altında bulunan şey ise eğer kullanıcı uygulamanın güncelleme kısmından bir yetim etiket için dosya açarsa **OUTPUT** listesinin içinde etiketler yeniden yazdırılıyor ve listede bulunan yetim etiketlerden .txt uzantılı dosyası açılmış olanlar yetim etiketler listesinden siliniyor. Buna ek olarak güncelleme kısmından eğer kullanıcı bir etiketi

değiştirirse güncel listede etiket ismi değişmiş bir şekilde  
gözüküyor. Ve bu sayede uygulamanın çıktısı güncel bir  
şekilde kalıyor. Kullanıcı istediği zaman **OUTPUT** dosyası  
içerisinde yetim etiketlere veya normal etiketlere ulaşabiliyor.  
Ve dosyalar içerisinde kaç defa bulunduğunu görebiliyor.

## YETİM ETİKETLER

Dosyalarımızın içerisinde bulunan 2 çeşit etiket vardır. Bunlardan bir tanesi olan **Yetim Etiketleri** normal etiketlerden ayrıran özellik, **Yetim Etiketler** bir çeşit etikettir fakat .txt uzantılı bir dosyaya sahip değildir. Fakat kullanıcı isterse bu etiketler içinde .txt uzantılı dosya oluşturma imkanına sahiptir.

## İSTENEN ETİKETLER

**İstenen Etiketler** .txt uzantılı dosyaya sahiptir fakat dosyaların içerisinde etiket olarak bulunmazlar.

## DOSYA YOLLARININ ALINMASI

Dosya yollarını alırken öncelikle system komut satırı ile klasörümüzün içine girerek alt klasörlerin adını bir tane .txt uzantılı dosyanın içine gönderiyoruz. Ardından bu dosyadan isimleri sırayla alarak alt klasörler açılıyor. Ve daha sonrasında bu klasörlerin içindeki .txt uzantılı dosyaların isimleri tekrardan başka bir .txt uzantılı dosyanın içine gönderiliyor. Bu isimlerde klasör isimlerinde olduğu gibi sırayla .txt uzantılı dosyalar açılıyor. Ve bu işlem sayesinde yapmak istediğimiz işlemleri gerçekleştirmek için dosyalarımıza erişmiş oluyoruz.

## DENEYSEL SONUÇLAR

```
void klasorgazme(char *dosyayolu[250], int *i)
{
    system("cd universe & dir \"/ad/b/d/o:dosyaislist.txt");
    char dosyaisimleri = fopen("./universe/dosyaislist.txt", "r");
    char satir[1500];
    char dersattir[1500];
    char dosyalar[150], dizi[100], dizi2[100];
    *i = 0;
    while (fgets(satir, 1500, dosyaisimleri) != NULL)
    {
        satir[strlen(satir) - 1] = '\0';
        strcpy(diz1, "cd ./\universe\");
        strcpy(dizi2, "DIR *.TXT/b/o:ns./derslist2.txt");
        strcat(dizi, satir);
        strcat(dizi, dizi2);
        system(dizi);
        FILE *dersisimleri = fopen("C:\Users\tunang\Desktop\ProLab\universe\derslist2.txt", "r");
        while (fgets(dersattir, 1500, dersisimleri) != NULL)
        {
            dersattir[strlen(dersattir) - 1] = '\0';
            sprintf(dosyalar, "C:\Users\tunang\Desktop\ProLab\universe\%s\%s", satir, dersattir);
            dosyayolu[*i++] = strdup(dosyalar);
        }
    }
}
```

## Klasörlerin İçerisinde Gezme Fonksiyon

Tüm uygulamanın çalışmasını sağlayan temelde bu fonksiyondur. Bu fonksiyon sayesinde tüm klasörler ve dosyalar arasında dolaşmış oluyoruz.

```
void istenemetik(int from(char *aramadosyasi[100],char *istenemetik[100],char *unig_isteler[100],int unig_index,int i,int pp,
int domus = 0;
int pp = 0;
for (int i = 0; i < k; i++)
    aramadosyasi[i] = strlen(aramadosyasi[i]) - 5) * '\0';

for (int i = 0; i < k; i++)
{
    domus = 0;
    for (int j = 0; j < unig_index; j++)
    {
        if (strcmp(unig_isteler[j], aramadosyasi[i]))
        {
            domus = 1;
        }
    }
    if (domus == 0)
    {
        istenemetik[pp] = aramadosyasi[i];
        (pp)++;
    }
    *p=pp;
}
```

## İstenen Etiket Bulma Fonksiyonu

**İstenen Etiketler** bu fonksiyon sayesinde bulunmakta. Ve main fonksiyon içerisinde ekrana yazdırılmaktadır.

```
void yetimlik(char *unik_inisler[100], int unik_index, char *aramadosyasi[150], char *yetimliket[100], int i, int k, int *donus)
{
    char satir[1500], derasatir[1500], dizil[150], dizil2[150];
    FILE *yetimliketVolu = fopen("C:\\Universite\\dosyalist.txt", "r");
    int y = 0;
    int satirL;
    while (!fgets(satir, 1500, yetimliketVolu) != NULL)
    {
        satir[strlen(satir) - 1] = '\0';
        strcpy(dizil, "C:\\Universite\\");
        strcpy(dizil2, "B DİR " + "TUTULAN" + derasatir + ".txt");
        strcat(dizil, satir);
        strcat(dizil, dizil2);
        system(dizil);
        FILE *yetimlikdosyasilari = fopen("C:\\Users\\tunagi\\Desktop\\Probab\\Universite\\derslist2.txt", "r");
        while (!fgets(derasatir, 1500, yetimlikdosyasilari) != NULL)
        {
            aramosyasi[y] = strdup(derasatir);
            {
                D[y];
                k++;
            }
            k++;
            y++;
        }
        k++;
        yetimliket[y] = unik_inisler[i];
    }
    *donus = 0;
    for (int j = 0; j < k; j++)
    {
        if (strcmp(aramodosyasi[j], unik_inisler[i]))
        {
            *donus = 1;
        }
        if (*donus == 0)
        {
            yetimliket[a] = unik_inisler[i];
            a++;
            i++;
        }
    }
}
```

## Yetim Etiket Bulma Fonksiyonu

**Yetim Etiketler** bu fonksiyon sayesinde bulunmakta. Ve main fonksiyon içerisinde ekrana yazdırılmaktadır.

```
void output(char *unig_isiner[100],char *yetimetiket[100],int unig_sayisi[100],int unig_index,int i,int sayac){
FILE *openfilepath; //fopen("C:\Users\tung\Desktop\ProLab\universite\output.txt","w");
fprintf(openfilepath,"Etiket isinleritv(tvtvtekrarsayis\n");
while(++sayac < unig_index)
{
    fprintf(openfilepath,"X-40s\t%d\n", unig_isiner[sayac], unig_sayisi[sayac]);
}
fclose(openfilepath);
openfilepath = fopen("C:\Users\tung\Desktop\ProLab\universite\output.txt", "a");
fprintf(openfilepath, "\nYeni Etiketler\n");
sayac = -1;
while (++sayac < 1)
{
    fprintf(openfilepath, "X-40s\n", yetimetiket[sayac]);
}
fclose(openfilepath);
}

void outputguncel(char *unig_isiner[100],char *yetimetiket[100],int unig_sayisi[100],int unig_index,int i,int sayac){
FILE *openfilepath; //fopen("C:\Users\tung\Desktop\ProLab\universite\output.txt", "a");
fprintf(openfilepath, "\nGuncel Isiler\n");
fprintf(openfilepath, "Etiket isinleritv(tvtvtekrarsayis\n");
while(++sayac < unig_index)
{
    fprintf(openfilepath, "X-40s\t%d\n", unig_isiner[sayac], unig_sayisi[sayac]);
}
fclose(openfilepath);
openfilepath = fopen("C:\Users\tung\Desktop\ProLab\universite\output.txt", "a");
fprintf(openfilepath, "\nYeni Etiketler\n");
sayac = -1;
while (++sayac < 1)
{
    fprintf(openfilepath, "X-40s\n", yetimetiket[sayac]);
}
fclose(openfilepath);
}
```

## OUTPUT Fonksiyonu

**OUTPUT** dosyasının oluşması ve eğer uygulamada güncelleme yapılırsa, güncel **OUTPUT** bu iki fonksiyon sayesinde yapılmaktadır.

```
void etiketlerVeriKontrolu(char *isimler[100],char *wmg_isimler[100],int *wmg_index,int *wmg_sayisi[100],int index){
    for (int i = 0; i < index; i++) // isimlerin içinde etiketler indeksini arastırarak satır 04
    {
        int gecici_sayisi = 0;
        int ulu_wmg_index;
        for (int j = 0; j < index && ctrl_str(isimler, isimler[i], 1); j++) // kontrol yapısını anlayabilişiniz onun için onu açıklıyorum
            // isimler dizisinde etiketleri tek tek arıyor ve ayırtmaya denk gelinceye kadar sayısını bir artırıyor
        {
            if (ft_strcmp(isimler[i], isimler[j]))
                gecici_sayisi++;
        }
        if (gecici_sayisi != 0)
        {
            wmg_isimler[ulw] = strcpy(isimler[i]);
            wmg_sayisi[ulw] = gecici_sayisi;
            ulw++;
        }
        *wmg_index=ulw;
    }
}
```

### Etiket Tekrar Sayısını Bulma Fonksiyonu

```

if (count == 0)
{
    printf("\nGizliHarfLeri:\n");
    for (int i = 0; i < uzm_kodlamayla; i++)
    {
        printf("%d\n",uzm_kodlamayla[i]);
    }

    printf("\nGizliHarf AramaK istedigiler kelime veya etiket giriniz:\n");
    char aramayi;
    scanf("%s",&aramayi);
    aramayi=strlen(aramayi)-1;
    int kontrol = 0;
    for (int i = 0; i < satirsayisi; i++)
    {
        char satir[uzm_kodlamayla];
        char cumele[uzm_kodlamayla];
        FILE *p = fopen("GizliHarfLeri.txt","r");
        int satirsayisi2 = 1;
        while (fgets(cumele, uzm_kodlamayla, p))
        {
            char *yaz = strstr(cumele, aramayi);
            while (ret != NULL)
            {
                int count = 0;
                int basindex = ret - cumele;
                int sonindex = strstr(cumele, aramayi) - cumele;
                if (basindex < sonindex)
                {
                    for (int i = 0; i < satirsayisi; i++)
                    {
                        if (aramayi[i] == " ")
                        {
                            continue;
                        }
                        if (count < 2)
                        {
                            printf("%s etiketleri ve %s dogussayisi %d.satirdahar %s\n", aramayi, derssatiri, satirsayisi);
                            continue;
                        }
                        else
                        {
                            printf("%s etiket degildir ve %s dogussayisi %d.satirdahar %s\n", aramayi, derssatiri, satirsayisi);
                            continue;
                        }
                    }
                }
                if (strlen(cumele[sonindex])>0)
                {
                    printf("%s etiket degildir ve %s dogussayisi %d.satirdahar %s\n", aramayi, derssatiri, satirsayisi);
                    continue;
                }
            }
            ret = strstr(ret + 1, aramayi);
        }
        printf("%s\n", satirsayisi);
        fclose(p);
    }
    if (kontrol == 0)
    {
        printf("%s herhangi bir yerde kelime olarak gecenokunulur.", aramayi);
    }
    printf("\nGizliHarfLeri:\n");
    for (int i = 0; i < uzm_kodlamayla; i++)
    {
        printf("%d\n",uzm_kodlamayla[i]);
    }
}

```

Bu fonksiyon sayesinde kullanıcının aramak istediği kelime etiket mi yoksa normal bir kelime mi ve istediği kelime hangi dosyaların içerisinde, hangi satırlarda geçtiğini bulmasına yarıyor.

```
char* etiketdegistirme(char* kopyala,char* degisecek_etiket,char* yeni_etiket,int *con)
{
    char* change_file;
    int i, cnt = 0;
    strcpy(degisecek_etiket,"[]");
    int yeniboyut = strlen(yeni_etiket);
    int eskiboyut = strlen(degisecek_etiket)-2;
    for (i = 0; kopyala[i] != '\0'; i++) {
        if (strstr(kopyala[i], degisecek_etiket) == kopyala[i]) {
            cnt++;
            *con--;
            i += eskiboyut ;
        }
    }
    change_file = (char*)malloc(i + cnt * (yeniboyut - eskiboyut + 1));
    i = 0;
    while (*kopyala) {
        if (strstr(kopyala, degisecek_etiket) == kopyala) {
            strcpy(change_file[i], yeni_etiket);
            i = yeniboyut;
            kopyala += eskiboyut ;
        }
        else change_file[i++] = *kopyala++;
    }
    change_file[i] = '\0';
    return change_file;
}
```

Bu fonksiyonumuz kullanıcının değiştirmek istediği etiket buluyor ve yerine yeni etiket koyuyor.

```
int dosyadigidegi(char *yenidosyaismi,char *eskidosya,char *dosyayolu[100],int boyut)
{
    for (int i = 0; i < strlen(yenidosyaismi); i++)
    {
        if (yenidosyaismi[i] == '.')
        {
            yenidosyaismi[i] = '_';
            strcat(yenidosyaismi, ".txt");
        }
        int a = 0;
        char *ret;
        yenidosyayolu=malloc(250);
        char *dosyayy,"dost";
        for (int i = 0; i < boyut; i++)
        {
            if ((dosyayy = strrchr(dosyayolu[i], '\\'))){
                strcpy(dost,dosyayy+1);
                dost[strlen(dost)-4]='\\0' ;
            }
            if ((ret = strstr(dosyayolu[i],eskidosya))){
                if ((ft_strstr(dost,eskidosya))=1)
                {
                    int b = ret - dosyayolu[i];
                    memset(yenidosyayolu,0,strlen(yenidosyayolu));
                    strcpy(yenidosyayolu, dosyayolu[i], b);
                    strcat(yenidosyayolu, yenidosyaismi);
                    rename(dosyayolu[i], yenidosyayolu);
                    strcpy(dosyayolu[i],yenidosyayolu);
                    printf("Dosya ismi basariyla degistirildi...");
                    a++;
                }
            }
        }
        if (a == 0)
            printf("Dosya ismi degistirilemedi");
    }
}
```

```
void tutmetiketler(char *dosyayolu[100],int txtsayisi,char *etiket[100],int *etiketsayisi)
{
    char *aranan = "[{}";
    char *aranan2 = "[]]";
    for (int i = 0; i < txtsayisi; i++)
    {
        char cumle[1000];
        FILE *fp = fopen(dosyayolu[i], "r");
        while (fgets(cumle, 1000, fp))
        {
            char *ret = strstr(cumle, aranan);
            char *ret2 = strstr(cumle, aranan2);
            while (ret != NULL && ret2 != NULL)
            {
                int cont = 0, cont1=0;
                char *etiket_ismi;
                int len = 0;
                while (ret[len + 2] != ']')
                {
                    len++;
                }
                etiket_ismi = malloc(len + 1);
                len = 0;
                while (ret[len + 2] != ']')
                {
                    etiket_ismi[len] = ret[len + 2];
                    len++;
                }
                etiket_ismi[len] = '\0';
                for (int i = 0; i < strlen(etiket_ismi); i++)
                {
                    if (etiket_ismi[i] == ' ')
                    {
                        cont++;
                    }
                    if (ispunct(etiket_ismi[i]) && etiket_ismi[i] != '_')
                    {
                        cont1++;
                    }
                }
                if (cont < 2 && cont1==0)
                {
                    etiketler[(*etiketsayisi)++] = strdup(etiket_ismi);
                }
                ret = strstr((ret) + 1, aranan);
                ret2 = strstr((ret2) + 1, aranan2);
            }
        }
        fclose(fp);
    }
}
```

Bu fonksiyon sayesinde dosyalar içerisinde geçen ve **OUT-PUT** içerisine yazılacak olan tüm etiketler bulunuyor.

Bu kısımda yazdığımız kodların nasıl çalıştığını ve menüde belirtilen işlemlerden etiket ve kelime arama ve güncelleme seçeneklerinin çalıştırıldıktan sonra ekran çıktılarının nasıl olduğu görülmektedir. Ek olarak OUTPUT.txt dosyasının nasıl olduğu da bu bölümde görülmektedir.

```

1-Kelime veya Etiket Arama
2-İçerik Güncelleme
3-Konsolu Temizleme
4-Kıyas Yapma
Lütfen yapmak istediğiniz işlemi seçiniz:1

Etiketler
tarih
ingilizce
türkçe
fizik
lineer cebir
matematik
bilgisayar laboratuvarı
programlama
kimya
teknik resim
bilgisayar laboratuvarı
Programlama
elektronik
türkçe
ProgramlamaLab
Programlama_1

Lütfen aramak istediğiniz kelime veya etiketi giriniz:programlama
programlama etikettir ve Bilgisayar Muhendisligi.txt dosyasında 9.satırdadır.
programlama etikettir ve bilgisayar laboratuvarı.txt dosyasında 4.satırdadır.
programlama etikettir ve bilgisayar laboratuvarı.txt dosyasında 5.satırdadır.
programlama etikettir ve programlama.txt dosyasında 2.satırdadır.

İstenen Etiketler
Bilgisayar Muhendisligi
Çevre Muhendisligi
Elektrik Muhendisligi
Endüstri Muhendisligi
Makine Muhendisligi
Mucak İnce

```

Etiket ve Kelime aramanın ekran çıktısı.

```
dersler
bilgisayar laboratu...
elektronik.txt
fizik.txt
ingilizce.txt
kimya.txt
lineer cebir.txt
matematik.txt
programlama.txt
tarrrr.txt
teknik resim.txt
turkce.txt
ogretim elemanlari
derslist2.txt
dosyalist.txt
output.txt
yetim etiketler.txt
derslist1.txt
dosyalist.txt
proje.c
proje.exe
NAHAT
1-Kelime veya Etiket Arama
2-Icerik Guncelleme
3-Konsolu Temizleme
4-Cikis Yapma
Lutfen yapmak istediginiz islemi seciniz:2
1-Etiket ve Dosya Adi Guncelleme
2-Yetim Etiket Dosyasi Acma
3-Dosya Adi Degistirme
Yapmak istediginiz islemi seciniz:1
Etiketler
Tarih
Ingilizce
Turkce
Fizik
Lineer cebir
matematik
Bilgisayar laboratuvarı
programlama
kimya
teknik resim
bilgisayar laboratuvarı
Programlama
elektronik
turkce
ProgramlamaLab
Programlama_I
Lutfen degistirmek istediginiz etiketi giriniz:tarih
Lutfen yeni etiketi giriniz:tarrrrr
Dosya ismi basariyla degistirildi...
Istenen Etiketler
Bilgisayar Muhendisligi
Cevre Muhendisligi
Elektrik Muhendisligi
Endustri Muhendisligi
Makine Muhendisligi
Burak Iner
```

## REFERENCES

- [1] <https://stackoverflow.com>
- [2] <https://www.bilgigunlugum.net>
- [3] <https://turkmuhendis.net>
- [4] <https://www.javatpoint.com>
- [5] <https://www.geeksforgeeks.org>
- [6] <https://www.tutorialspoint.com/index.htm>
- [7] <https://www.technopat.net>

## Etiket ve Dosya Adı Güncelleme

Güncelleme fonksiyonunun bir seçeneği olan Etiket ve Dosya Adı Güncellemenin ekran çıktısı.

```
Seç c:\Users\tunag\Desktop\ProLab\proje.exe
1-Kelime veya Etiket Arama
2-Icerik Guncelleme
3-Konsolu Temizleme
4-Cikis Yapma
Lutfen yapmak istediginiz islemi seciniz:2
1-Etiket ve Dosya Adi Guncelleme
2-Yetim Etiket Dosyasi Acma
3-Dosya Adi Degistirme
Yapmak istediginiz islemi seciniz:2
Yetim Etiketler
bilgisayar laboratuvarı
Programlama
turkce
ProgramlamaLab
Programlama_I
Lutfen dosyasını oluşturmak istediginiz yetim etiketi giriniz:Programlama
Yetim etiket dosyası oluşturuldu
```

## Yetim Etiket Dosyası Oluşturma

Yetim etiketler için .txt uzantılı dosya oluşturma ve o dosyanın içerisine şablon oluşturmaya yarayan fonksiyonun ekran çıktısı.

```
output.txt - Not Defteri
Dosya  Düzenle  Görünüm
Etiket isimleri  TekrarSayisi
Tarih  5
Ingilizce  4
Turkce  4
Fizik  4
Lineer cebir  4
matematik  1
Bilgisayar laboratuvarı  1
programlama  1
elektronik  1
turkce  1
ProgramlamaLab  1
Programlama_I  1
Yetim Etiketler
bilgisayar laboratuvarı
Programlama
turkce
ProgramlamaLab
Programlama_I
Guncel liste
Etiket isimleri  TekrarSayisi
Tarih  5
Ingilizce  4
Turkce  4
Fizik  4
Lineer cebir  4
matematik  1
Bilgisayar laboratuvarı  1
programlama  1
kimya  1
teknik resim  1
bilgisayar laboratuvarı  1
Programlama  1
elektronik  1
turkce  1
ProgramlamaLab  1
Programlama_I  1
Yetim Etiketler
bilgisayar laboratuvarı
Programlama
turkce
ProgramlamaLab
Programlama_I
```

## OUTPUT Çıktısı

OUTPUT fonksiyonunun .txt uzantılı dosyaya yazılmış görüntüsü.

## PROGRAM AKIŞ ŞEMASI

### A. *menu fonksiyonu:*

1-Konsolda arayüzü yazdırıyor.

### B. *ft\_strstr fonksiyonu:*

- 1-Aldığı iki değişkeni karakter karakter karşılaştır.
- 2-Kelime olarak aynı çıkıyorlarsa 1 değerini döndür.
- 3-Aynı çıkmıyorlarsa 0 değerini döndür.

### C. *ctrl\_str fonksiyonu:*

- 1-bir dizi, string ve int index parametre al.
- 2-dizinin index değerine kadar string olup olmadığını kontrol et.
- 2.1  
1-Eğer aynı string varsa 0 döndür.
- 2.2  
1-Eğer aynı string yoksa 1 döndür.

### D. *klasorgezme fonksiyonu:*

- 1-universite klasöründeki klasör isimlerini "dosyalist.txt" dosyasına yaz.
- 2-"dosyalist.txt" dosyasını satır satır okuyup klasör adlarını satir değişkenine al.
- 3-satır değişkenindeki klasörü aç.
- 4-klasördeki txt dosyalarını "derslist2.txt" dosyasına yaz.
- 5-"derslist2.txt" dosyasını satır satır okuyup txt dosya adlarını derssatir değişkenine al.
- 6-dosyalar değişkenine dosya uzantısı, satir ve derssatir değişkenlerini ekle.
- 7-dosyayolu dizisinin i değişkenine dosyalar değişkenini ata.

### E. *tumetiketler fonksiyonu:*

- 1-dosyayolu dizisini tek tek gez.
- 2-indexteki dosyayolunu okuma modunda aç.
- 3-dosyayı satır satır al.
- 4-ret değişkeni için satırda aranan değişkenini bul.
- 4.1  
1-Bulursa ret2 değişkeni için aranan2 değişkenini bul.
- 2-aranan2 değişkenine kadar len değişkenini artırıp kelime boyutunu bul.
- 3-etiket\_ismi değişkeninin boyutunu len+1 olarak ayarla.
- 4-etiket\_ismi değişkenine ret değişkenini ']' karakterini görene kadar ata.
- 5-etiket\_ismi değişkenini isimler dizisine ata.
- 6-ret ve ret2 değerlerini artırarak satır sonuna kadar aramaya devam et.
- 4.2  
1-Bulamazsa sonraki satıra geç.

### F. *etikettekrarkontrolu fonksiyonu:*

- 1-isimler dizisi eleman sayısı kadar döngüye gir.
- 2-bu döngünün içinde tekrardan isimler dizisi eleman sayısı kadar ve ctrl\_str doğru oldukça döngüye gir.
- 3-isimler dizisinde aynı elemanı buldukça gecissayisi değişkenini bir artır.
- 3.1  
1-gecissayisi 0'dan farklı ise
- 2-uniq\_isimler dizisine isimler elemanını ata.
- 3-uniq\_sayisi dizisine gecissayisini ata.
- 3.2  
1-baştaki döngüye dön.

### G. *yebulma fonksiyonu:*

- 1-"dosyalist.txt" dosyasını satır satır okuyup klasör adlarını satir değişkenine al.
- 2-satır değişkenindeki klasörü aç.
- 3-klasördeki txt dosyalarını "derslist2.txt" dosyasına yaz.
- 4-"derslist2.txt" dosyasını satır satır okuyup txt dosya adlarını derssatir değişkenine al.
- 5-derssatir değişkenlerini aramadosyasi dizisine aktar.
- 6-etiket sayısı kadar kontrol sağla
- 7-"aramadosyasi" dizisindeki elemanlarla etiketleri karşılaştır.
- 7.1  
1-Bulursa donus değişkenine 1 değerini ata
- 7.2  
1-Bulamazsa etiket ismini "yetimetiket" dizisine ata.

### H. *output fonksiyonu:*

- 1-universite klasör yolunu vererek "output.txt" adında dosyayı yazma modunda aç.
- 2-uniq\_index sayısı kadar döngüye sok.
- 3-dosyaya etiket isimlerini ve tekrar sayılarını yazdır
- 4-dosyayı kapat.
- 5-"output.txt" dosyasını güncelleme modunda aç.
- 6-yetimetiket sayısı kadar döngüye sok.
- 7-dosyaya yetim etiketleri yazdır.
- 8-dosyayı kapat.

### I. *outputguncel fonksiyonu:*

- 1-universite klasör yolunu vererek "output.txt" adında dosyayı güncelleme modunda aç.
- 2-uniq\_index sayısı kadar döngüye sok.
- 3-dosyaya etiket isimlerini ve tekrar sayılarını yazdır.
- 4-dosyayı kapat.
- 5-"output.txt" dosyasını güncelleme modunda aç.
- 6-yetimetiket sayısı kadar döngüye sok.
- 7-dosyaya yetim etiketleri yazdır.
- 8-dosyayı kapat.

#### **J. istenenetiket fonksiyonu:**

- 1-aramadosyasi sayısı kadar döngüye sok.
- 2-dizi elemanlarının sondan 5.karakterlerine bitirme karakterini ekle
- 3-tekrar aramadosyasi sayısı kadar döngüye sok.
- 4-bu döngü içinde uniq\_isimler sayısı kadar döngüye sok.
- 5-aramadosyasi elemanları ile uniq\_isimler elemanlarını karşılaştır.
- 5.1  
1-Bulursa donus degerine 1 ata.
- 5.2  
1-Bulamazsa donus degeri değiştirme.
- 6-Donus değeri 0 ise aramadosyasi elemanını istenenetiket dizisine ata

#### **K. fdosyaadidegisme fonksiyonu:**

- 1-yenidosyaismi string karakterlerini tek tek kontrol et
- 2-Boşluk karakteri ' ' bulunursa '\_' karakteri ile değiştir.
- 3-yenidosyaismi stringinin sonuna ".txt" ekle.
- 4-dosyayolu dizisi eleman sayısı kadar donguye gir.
- 5-dosyayolu dizisi elemanında ters slash karakterini sondan ara.
- 6-bulunduğu noktadan itibaren dost stringine aktar.
- 7-dost stringinin sondan 4.karakterine bitiş karakteri koy.
- 8-eski dosya ismi ile dosyayolu elemanı karşılaştı.
- 8.1  
1-Bulunursa ft\_strstr kontrolü sağla.
- 2-Eğer kelime olarak da aynıysa dosya ismi değiştir.
- 8.2  
1-ft\_strstr kontrolü sağlanmıyorsa dosya ismi değiştirme.

#### **L. etiketdegistirme fonksiyonu:**

- 1-dosyanın içini yazdığımız stringi harf harf gez.
- 2-stringin içinde degisecek\_etiketi ara
- 3-buldukça cnt değişkenini artır.
- 4-değişecek etiket ve yeni etiket arasındaki farkı bulup cnt değişkeniyle çarpıp açılacak alan farkını bul.
- 5-değişecek yere gelince yeni etiketi kopyala.
- 6-değişim yapılmayacak yerlerde dizileri eşitle.

#### **M. main fonksiyonu:**

- 1-Seçim=1** ise
- 1.1-Aranacak kelimeyi gir.
- 1.2-dosyayolu dizisindeki dosyaları okuma modunda aç.
- 1.3-dosyaları satır satır oku.
- 1.4-aranan kelimeyi görünce etiket kontrolü yap.
- 1.5-etiketse dosya adını ve satır numarasını ekrana bastır.
- 1.6-etiket değilse dosya adını ve satır numarasını bastır.
- 1.7-eğer dosyaların hiçbirinde geçmiyorsa herhangi bir yerde geçmediğini bastır.
- 2-Seçim=2** ise
- 2.1-Seçenek=1** ise
- 2.1.1-Değişecek etiketi gir.
- 2.1.2-Yeni etiketi gir.
- 2.1.3-dosyayolu dizisindeki dosyaları okuma modunda aç.

2.1.4-dosyanın boyutunu bul ve kopyala stringinin boyutunu ayarla.

2.1.5-dosyanın içini "kopyala" stringine yaz.

2.1.6-alınan ve oluşturulan parametreleri etiketdegistirme fonksiyonuna yolla.

2.1.7-dosyayolunu yazma modunda aç ve değiştirilmiş stringi içine yaz.

2.1.8-etikitle aynı isimde txt dosyası varsa adını değiştir.

**2.2-Seçenek=2** ise

2.2.1-yebulma fonksiyonunu çalıştır.

2.2.2-yetim etiketleri ekranda göster.

2.2.3-yetim etiketlerden birini seç.

2.2.4-dosyayolunu verip yetim etiket adında dosya oluştur.

2.2.5-dosyanın içine gerekenleri yaz.

**2.3-Seçenek=3** ise

2.3.1-değiştirilecek dosya adını gir.

2.3.2-yeni dosya adını gir.

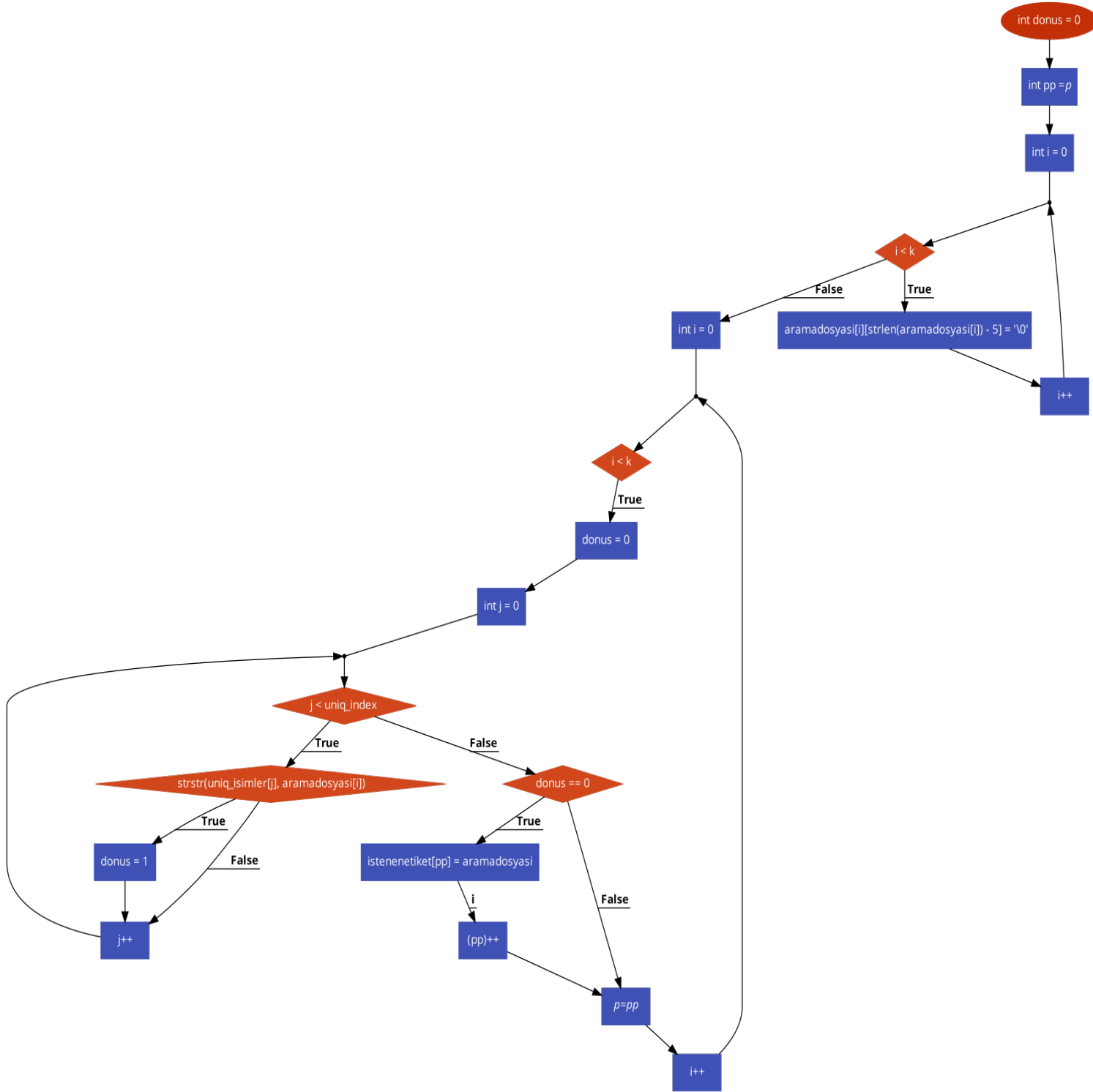
2.3.3-dosyadidegisme fonksiyonuna parametreleri yollayıp çalıştır.

**3-Seçim=3** ise

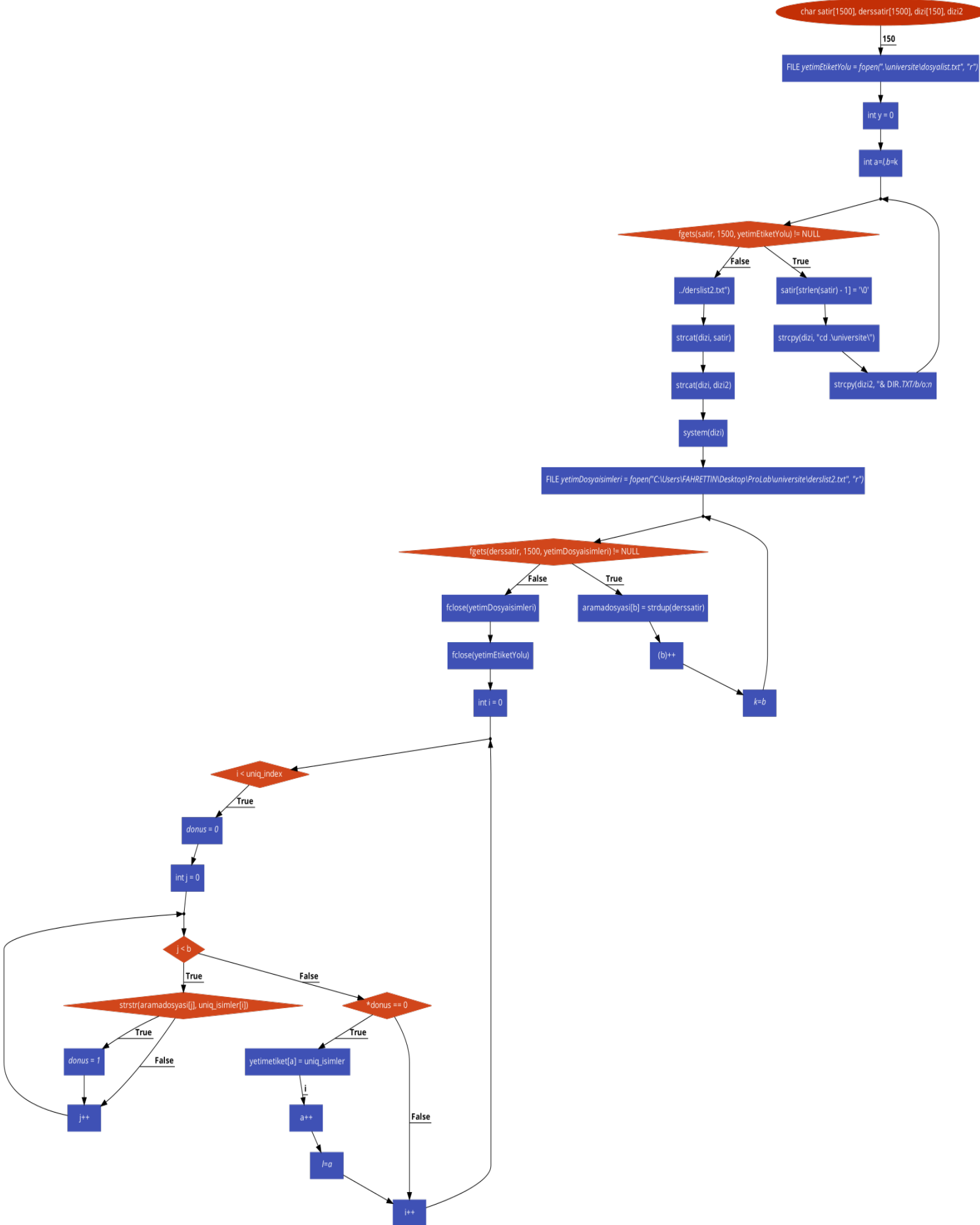
1-Konsolu temizleme komutunu çalıştır.

**4-Seçim=4** ise

1-Uygulamadan çıkış yap.



**İstenen Etiket Akış Şeması**



**Yetim Etiket Akış Şeması**