

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuvarı I

Bağlı Liste ile Kelime Sayma

Taha Uz/Burak Emir Ayrancı

maskedn_2000@hotmail.com[Taha]/190202014@kocaeli.edu.tr[Emir]

Projenin Özeti:

Programlama Laboratuvarı dersinin 3.Projesi olarak bizden “Bağlı Liste ile Kelime Sayma” adlı uygulama geliştirmemiz beklenmektedir.

Biz proje için C programlama dilini ve CodeBlocks geliştirme ortamını seçtik.

Proje dökümanından bizden beklenen “.txt” dosyasından girilen bir metnin içinde geçen kelimeleri ya da noktalama işaretlerini teker teker bağlı listenin içinde tutucak bir yapı oluşturmamız beklenmektedir.

İlk olarak “metin.txt” dosyasından girilen metindeki kelimeleri teker teker okuyup “struct_kelime_t” adlı bir yapıya atıyor o yapıda metindeki kelimeler tek tek “veri” adlı “char” tipindeki değişkende tutuluyor.

```
typedef struct struct_kelime_t
{
    char *veri;
    int kelimeSayaci;
    struct struct_kelime_t* sonraki;
} kelime_t;
```

Eğer aynı kelime denk gelirse bağlı listeye farkı bir değişkenmiş gibi eklemiyor onun yerine aynı değişkende “kelimeSayaci” adlı “int” tipindeki değişkende sayacı “+1” arttırarak bağlı listede tutuluyor.Metni okuma evresi bittiğinde diğer aşama olan bağlı listeye “Başa Ekle”, “Araya Ekle”, “Sona Ekle” gibi programın menüsündeki seçeneklerle bağlı listeye kelimeler eklenebiliyor.

1.GİRİŞ

Proje için C programlama dilini ve CodeBlocks geliştirme ortamını kullandık. C programlama dili; taşınabilirliği ve ifade gücü çok yüksek, okunabilirlik özelliği güçlü olan esnek bir dildir. CodeBlocks platformu; özgür açık kaynak kodlu bir C/C++ tümleşik geliştirme ortamıdır. wxWidgets tabanlı tamamen özelleştirilebilir arabirimiyle gelişmiş plugin desteğiyle birçok yardımcı fonksiyon sunan bir geliştirme platformudur.

2.TEMEL BİLGİLER

Projemiz; dosyadan karakter okuyup bağlı listeye göndermek , bağlı listeyi kelimeleri sıralarken kelimenin metinde kullanılan sıklığına göre sıralarken büyükten küçüğe sıralama,bağlı listeye basa ekleme ,bağlı listeye araya ekleme,bağlı listenin sonuna ekleme olmak üzere 5 temel bölüme ayırdık. Bu projede “stdio.h,stdlib.h,string.h,ctype.h,locale.h,stdbool.h” kütüphanelerini kullandık.

2.1.Dosyadan Karakter Okumak ve Bağlı Listede Toplamak

Dosyadan kelimeleri karakter karakter okuyor dosyadan alınan bu karakterler “ch”adlı “int” tipinde değişkene o karakterlerin ASCII tablosundaki sayı değerlerini atıyor.İlk kontrol karakter bosluk(' ') ya da enter('\n') karakterine eşit mi değil mi onu kontrol ediyor. Bu kontroller ile kelimeleri ayırıp oraya kadar okunan karakteri kelime olarak “kelimeEkle” fonksiyonuna yolluyor.

```
int ch = 0, kelime = 1, k = 0;
char kelime_tut[100];

while ((ch = fgetc(myFile)) != EOF)
{
    if (ch == ' ' || ch == '\n')
    {
        if ( kelime == 1 )
        {
            kelime = 0;
            kelime_tut[k] = '\0';
            mykelime_t = kelimeEkle(mykelime_t, kelime_tut);
            k=0;
        }
    }
    else
    {
        kelime = 1;
        kelime_tut[k] = ch;
        k++;
    }
}
if (kelime == 1)
{
    kelime_tut[k] = '\0';
    mykelime_t = kelimeEkle(mykelime_t, kelime_tut);
}
```

Bu fonksiyonda alınan bu karakterler “kelime” adlı “char” tipi kelimeye yolluyor ve main fonksiyonun içinde tanımlanan “mykelime_t”

adlı “struct” tipindeki değişkeni de “kelimeEkle” fonksiyonuna yolluyor. Bu fonksiyonda ilk olarak “kelimeListe” bağlı listesinde gelen kelime var mı yok mu ona bakılıyor . Varsa kelimenin sayacı “+1” arttırılıyor. Böylece aynı kelime bağlı listede olmuyor onun yerine kaç kere metinde tekrar ettiğini görebiliyoruz.

```
while (temp != NULL)
{
    if (strcmp(temp->veri, kelime) == 0)
    {
        kontrol=true;
        temp->kelimeSayaci = temp->kelimeSayaci+1;
        return kelimeListe;
    }
    else
    {
        temp = temp->sonraki;
    }
}
```

İlk koşulu geçtikten sonra ikinci koşula geliyoruz. Bu koşul da bu listede eğer ilk koşuldan geçip yeni bir kelime olduğu kanıtlanırsa bu ikinci koşula girme hakkı kazanıyor.Böylece bu koşulun içinde “yeniKelime” adlı “struct” tipindeki değişkene “bagilListe_kelime_t” adlı fonksiyona kelime yollanarak bir düğüm oluşturuyor ve burdan veriler geri dönüp bu verileri

“yeniKelime->sonraki = kelimeListe” diyerek yeni kelimemizi bağlı listemize eklemiş oluyoruz.

```
if (kontrol==false)
{
    kelime_t* yeniKelime = bagilListe_kelime_t(kelime);
    if (NULL != yeniKelime)
    {
        yeniKelime->sonraki = kelimeListe;
    }
    return yeniKelime;
}
```

Böylece dosyadan okunan kelime bitene kadar kelimeleri bu döngüde bağlı listeye topluyor. “bagilListe_kelime_t” fonksiyonunda yeni kelimenin verilerini taşıdığı düğümü oluştururken “yeniKelime->sonraki” de her zaman “NULL” olacak şekilde ayarlanıyor.

```

kelime_t* bagilListe_kelime_t(char* kelime)
{
    int uzunluk = strlen(kelime);

    kelime_t* yeniKelime = malloc(sizeof(kelime_t));
    if (NULL != yeniKelime)
    {
        yeniKelime->veri = (char *)malloc((uzunluk)+1);
        strcpy(yeniKelime->veri, kelime);
        yeniKelime->veri[uzunluk]='\0';

        yeniKelime->kelimeSayaci = 1;

        yeniKelime->sonraki = NULL;
    }
    return yeniKelime;
}

```

Böylece listede ekle çıkar yaparken ona göre “yeniKelime->sonraki=AtanacakSıradakiDugum” yapısını kurduk.

2.2.Bağlı Listede Toplanan Kelimeleri Sıraya Sokmak

Bağlı listeye metindeki tüm kelimeleri atadıktan sonra sıralama kısmına geliyoruz.Bu işlemde Kabarcık Sıralama(Bubble Sort) sistemini kullanıyoruz.Bu sıralamada bağlı listenin geçicisini tutan “temp” adlı değişkene atıyoruz böylece döngüyü sağlarken listenin yapısını bozulmamasını sağlıyoruz.Bu döngünün içinde ilk olarak bir koşul durumu var bu koşul bağlı listedeki ilk eleman bir sonraki elemanın “kelimeSayaci” adlı “int” değerlerini karşılaştırıyor ve böylece kelime büyüklüğünü kontrol ediyor eğer büyükse koşul durumu sağlanıp koşul durumunun içindeki “swap” işlemi gerçekleşiyor eğer koşul sağlanmaz ise bir sonraki elemana geçiş yapıyor.Koşul durumunda düğümlerin “veri” adlı değişkeni ve “kelimeSayaci” adlı değişkeni arasında takas(swap) yapılıyor.Böylece bu işlemlerle bağlı listeyi belirli bir sıraya sokmuş oluyoruz.

```

while(temp->sonraki!=NULL)
{
    if(temp->kelimeSayaci < temp->sonraki->kelimeSayaci)
    {
        swap=temp->sonraki->veri;
        swap_l=temp->sonraki->kelimeSayaci;
        temp->sonraki->veri=temp->veri;
        temp->sonraki->kelimeSayaci=temp->kelimeSayaci;
        temp->veri=swap;
        temp->kelimeSayaci = swap_l;
    }
    temp=temp->sonraki;
}

```

2.3.Bağlı Listenin Başına Kelime Ekleme

Bağlı listenin başına kelime eklerken ilk olarak kullanıcının eklemek istediği kelimeyi “char” tipinde “kullanici_kelime” adlı değişkende tutuyoruz.Sonra kullanıcının aldığımız kelimeyi ve bağlı listeyi “basaKelimeEkle” adlı fonksiyona parametre olarak yolluyoruz.

```

char kullanici_kelime[20];
printf("Bagli Listeye Eklieceginiz Kelimeyi Giriniz : ");
scanf("%s",&kullanici_kelime);
mykelime_t = basaKelimeEkle(mykelime_t, kullanici_kelime);

```

Bu fonksiyonda ilk olarak kelimenin listede var olup olmadığını kontrol ediyoruz. Eğer listede bu kelime var ise sadece bağlı listede kelimeyi bulup kelimenin “kelimeSayaci” değişkenini “+1” arttırıyoruz ve kullanıcıya “Girdiginiz Kelimenin Aynisi Bagil Listede oldugu icin kelimeniz araya eklenemedi sadece kelimenin sayaci arttirildi.” Uyarısını yollayarak kullanıcıyı bilgilendiriyoruz .

```

while (temp != NULL)
{
    if (strcmp(temp->veri, kelime) == 0)
    {
        kontrol=true;
        temp->kelimeSayaci = temp->kelimeSayaci+1;
        printf("Girdiginiz Kelimenin Aynisi Bagli Listede oldugu icin kelimeniz araya eklenemedi sadece kelimenin sayaci arttirildi. \n");
        return kelimeListe;
    }
    else
    {
        temp = temp->sonraki;
    }
}

```

Eğer kelime listede yoksa yani kelime yeni bir kelimeyse “yeniKelime” adlı düğümü oluşturup bu düğüme girilen kelimeyi attıyoruz böylece “yeniKelime” adlı değişkende “veri” değişkeninde “kelime” değişkeni tutuluyor. Böylece “yeniKelime” adlı değişkeni “yeniKelime->sonraki=kelimeListe” diyip böylece kelimeyi listeye eklemiş olup “basaKelimeEkle” fonksiyonunda “return yeniKelime” diyerek güncel listeyi “mykelime_t” adlı değişkene yollayarak “main” içindeki listeyi güncelliyoruz.

```

if (kontrol==false)
{
    kelime_t* yeniKelime = bagilListe_kelime_t(kelime);

    yeniKelime->sonraki=kelimeListe;
    printf("Kelime[%s] bagli listenin basina eklendi.\n", kelime);

    return yeniKelime;
}

```

Böylelikle bağlı listenin başına kullanıcının girdiği kelime eklenmiş oluyor.

2.4.Bağlı Listenin Sonuna Kelime Ekleme

Bağlı listenin sonuna kelime eklerken ilk olarak kullanıcıdan eklemek istediği kelimeyi “char” tipinde “kullanici_kelime” adlı değişkende tutuyoruz.Sonra kullanıcıdan aldığımız kelimeyi ve bağlı listeyi “SonaKelimeEkle” adlı fonksiyona parametre olarak yolluyoruz.

```
char kullanici_kelime[20];
printf("Bağlı Listeye Eklieceğiniz Kelimeyi Giriniz : ");
scanf("%s",&kullanici_kelime);
mykelime_t = SonaKelimeEkle(mykelime_t, kullanici_kelime);
```

Bu fonksiyonda ilk olarak kelimenin listede var olup olmadığını kontrol ediyoruz. Eğer listede bu kelime var ise sadece bağlı listede kelimeyi bulup kelimenin “kelimeSayaci” değişkenini “+1” arttırıyoruz ve kullanıcıya “Girdiginiz Kelimenin Aynisi Bagil Listede oldugu icin kelimeniz araya eklenemedi sadece kelimenin sayaci arttirildi.” Uyarısını yollayarak kullanıcıyı bilgilendiriyoruz.

```
while (temp->sonraki == NULL)
{
    if (strcmp(temp->veri, kelime) == 0)
    {
        kontrol_teme:
        temp->kelimeSayaci = temp->kelimeSayaci+1;
        printf("Girdiginiz Kelimenin Aynisi Bagil Listede oldugu icin kelimeniz araya eklenemedi sadece kelimenin sayaci arttirildi. (%d)", temp->kelimeSayaci);
        return kelimeListe;
    }
    else
    {
        temp = temp->sonraki;
    }
}
if (strcmp(temp->veri, kelime) == 0)
{
    kontrol_teme:
    temp->kelimeSayaci = temp->kelimeSayaci+1;
    printf("Girdiginiz Kelimenin Aynisi Bagil Listede oldugu icin kelimeniz araya eklenemedi sadece kelimenin sayaci arttirildi. (%d)", temp->kelimeSayaci);
    return kelimeListe;
}
```

Eğer kelime listede yoksa yani kelime yeni bir kelimeyse “yeniKelime” adlı düğümü oluşturup

```
int uzunluk = strlen(kelime);

kelime_t* yeniKelime = malloc(sizeof(kelime_t));
if (NULL != yeniKelime)
{
    yeniKelime->veri = (char *)malloc((uzunluk)+1);
    strcpy(yeniKelime->veri, kelime);
    yeniKelime->veri[uzunluk]='\0';

    yeniKelime->kelimeSayaci = 1;

    yeniKelime->sonraki = NULL;
}
return yeniKelime;
```

bu düğüme girilen kelimeyi atıyoruz böylece “yeniKelime” adlı değişkende “veri” değişkeninde “kelime” değişkeni tutuluyor.Oluşturulan bu düğümü önceden döngüde döndürdüğümüz ve listenin son elemanına gelen “temp” adlı değişkenin “temp->sonraki” düğümü “NULL” değerine eşit olmuş oluyor ve böylece listenin sonuna geldiğimizi anlıyoruz. Kullanıcının girdiği değerleri saklayan düğümü “temp->sonraki=yeniKelime” diyerek yeni kelimeyi listenin sonuna eklemiş oluyoruz.

```
kelime_t* yeniKelime = bagilListe_kelime_t(kelime);
temp->sonraki = yeniKelime;
printf("Kelime[%s] bagli listenin sonuna eklendi.\n", kelime);
return kelimeListe;
```

Listeyi güncellediğimiz için yeni listeyi “return kelimeListe” diyerek “main” fonksiyonuna yollamış oluyoruz.

2.5.Bağlı Listede Araya Kelime Ekleme

Bağlı listenin arasına kelime eklerken ilk olarak kullanıcıdan eklemek istediği kelimeyi “char” tipinde “kullanici_kelime” adlı değişkende tutuyoruz.İkinci olarak da kullanıcıdan hangi kelimedenden sonra eklemek istediğini sorup girilen yanıtı “kullanici_kelime_sec” adlı “char” tipindeki değişkene atarak “ArayaKelimeEkle” fonksiyonuna kullanıcı tarafından girilen bu iki değişkeni parametre olarak yolluyoruz.

```
char kullanici_kelime[20];
char kullanici_kelime_sec[20];
printf("Bağlı Listeye Eklieceğiniz Kelimeyi Giriniz : ");
scanf("%s",&kullanici_kelime);
printf("Bağlı Listeye Hangi Kelimedenden Sonra Eklesin : ");
scanf("%s",&kullanici_kelime_sec);
mykelime_t = ArayaKelimeEkle(mykelime_t, kullanici_kelime,kullanici_kelime_sec);
```

Sona “ArayaKelimeEkle” adlı fonksiyonda işlemlere başlıyoruz.Öncelikli “kullanici_kelime_sec” adlı değişken bağlı listede olup olmadığını kontrol ediyor eğer yoksa direkt bağlı listede hiçbir şeyi

değiştirmeyip listeyi geri yolluyor bu hatayı da kullanıcıya “Normalde Bagli Listede Olan Girdiginiz Kelimeyi Yanlis Girdiniz Bagli Listede Yok!” uyarısıyla bilgilendiriyoruz.

```
while (temp_2 != NULL)
{
    if (strcmp(temp_2->veri, ara) == 0)
    {
        buldu=1;
        break;
    }

    temp_2 = temp_2->sonraki;
}

if(buldu == 0)
{
    printf("Normalde Bagli Listede Olan Girdiginiz Kelimeyi Yanlis Girdiniz Bagli Listede Yok!\n");
    return kelimeListe;
}
```

Sonra listeye eklenmesi için kullanıcının girdiği kelimeyi bağlı listede olup olmadığını kontrol eder. Eğer varsa o kelimenin “kelimeSayaci” adlı değişkenini “+1”arttırıp kullanıcıya "Girdiginiz Kelimenin Aynisi Bagli Listede olduğu için kelimeniz araya eklenemedi sadece kelimenin sayaci arttirildi.” Uyarıyı yaparak fonksiyon listeyi değiştirmeyip geri yollar.

```
while (temp != NULL)
{
    if (strcmp(temp->veri, kelime) == 0)
    {
        kontrol=true;
        temp->kelimeSayaci = temp->kelimeSayaci+1;
        printf("Girdiginiz Kelimenin Aynisi Bagli Listede olduğu için kelimeniz araya eklenemedi sadece kelimenin sayaci arttirildi. \n");
        return kelimeListe;
    }
    else
    {
        temp = temp->sonraki;
    }
}
```

Eğer kullanıcının girdiği kelime bağlı listede yoksa hemen yeni düğüm oluşturulur “yeniKelime” adında ve buraya kullanıcıdan alınan veriler girilir. Sonra “eskiList_Temp” adlı değişkene “temp_2->sonraki” adlı değişken atanır. “temp_2->sonraki = yeniKelime” diyerek kullanıcının şu kelimedden sonra ekle dediği kelimenin sonrasına eklenir.”yeniKelime->sonraki = eskiList_Temp” yaparak düğümün listenin “temp_2” döngüsünde kelimenin bulunduğu yerden devam eden kelimeleri

“yeniKelime” düğümünün sonrasına ekleyerek bütünlüğü bozmadan kelimeyi araya eklemiş oluruz.

```
kelime_t* yeniKelime = bagliListe_kelime_t(kelime);
eskiList_Temp=temp_2->sonraki;
temp_2->sonraki=yeniKelime;
yeniKelime->sonraki=eskiList_Temp;
printf("Kelime[%s] bagli listenin [%s] kelimesinin sonrasina eklendi.\n", kelime, ara);
```

3.DENEYSEL SONUÇLAR

```
*****
BAGLI LİSTE İLE KELİME SAYMA PROGRAMINA HOŞGELDİNİZ
*****
-----ANA MENU-----
*****
1-)Bagli Listeyi Goster
2-)Bagli Listenin Basina Kelime Ekle
3-)Bagli Listenin Arasina Kelime Ekle
4-)Bagli Listenin Sonuna Kelime Ekle
5-)Bagli Listeyi Buyukten Kucuge Dogru Sirala
6-)Programdan Cikis
*****
Lutfen Deger Giriniz [1-6 arasi rakam giriniz!]:
```

```
*****
Lutfen Deger Giriniz [1-6 arasi rakam giriniz!]: 2
*****
Bagli Listeye Ekliceginiz Kelimeyi Giriniz : Elma
Kelime[Elma] bagli listenin basina eklendi.
*****
```

```
*****
-----BAGLI LİSTE-----
*****
1-)Elma --> 1
2-). --> 5
3-)* --> 5
4-)daha --> 4
5-)kasim --> 4
6-)degismisti --> 3
7-)yillarinda --> 3
8-)2019 --> 3
9-)ve --> 3
10-)2016 --> 3
11-)once --> 3
```

```
*****
Lutfen Deger Giriniz [1-6 arasi rakam giriniz!]: 4
*****
Bagli Listeye Ekliceginiz Kelimeyi Giriniz : Kalem
Kelime[Kalem] bagli listenin sonuna eklendi.
*****
```

```
25-)yardimci --> 1
26-)2019'un --> 1
27-)goruluyor --> 1
28-)net --> 1
29-)yil --> 1
30-)gecen --> 1
31-)her --> 1
32-)etkileri --> 1
33-)isinmanin --> 1
34-)Kuresel --> 1
35-)Kalem --> 1
```

```

*****
Lutfen Deger Giriniz [1-6 arasi rakam giriniz!]: 3
*****
Bagli Listeye Ekliceğiniz Kelimeyi Giriniz : Bardak
Bagli Listeye Hangi Kelimeden Sonra Eklesin : rekor
Kelime[Bardak] bagli listenin [rekor] kelimesinin sonrasin
a eklendi.
*****

```

```

19-)oldu --> 1
20-)kirilmis --> 1
21-)rekor --> 1
22-)Bardak --> 1
23-)ayinda --> 1
24-)yilinin --> 1
25-)2020 --> 1
26-)ardindan --> 1

```

```

*****
-----ANA MENU-----
*****
1-)Bagli Listeyi Goster
2-)Bagli Listenin Basina Kelime Ekle
3-)Bagli Listenin Arasina Kelime Ekle
4-)Bagli Listenin Sonuna Kelime Ekle
5-)Bagli Listeyi Buyuktan Kucuge Dogru Sirala
6-)Programdan Cikis
*****
Lutfen Deger Giriniz [1-6 arasi rakam giriniz!]: 1
*****
-----BAGLI LESTE-----
*****
1-). --> 5
2-)* --> 5
3-)daha --> 4
4-)kasim --> 4
5-)degismisti --> 3
6-)yillarinda --> 3
7-)2019 --> 3
8-)ve --> 3
9-)2016 --> 3
10-)once --> 3
11-)rekoru --> 3
12-)sicak --> 3
13-)En --> 3
14-)... --> 1
15-)ntv.com.tr'de --> 1
16-)birazdan --> 1
17-)Detaylar --> 1
18-)oldu --> 1
19-)kirilmis --> 1
20-)rekor --> 1
21-)ayinda --> 1
22-)yilinin --> 1
23-)2020 --> 1
24-)ardindan --> 1
25-)2019'un --> 1
26-)goruluyor --> 1
27-)net --> 1
28-)yil --> 1
29-)gecen --> 1
30-)her --> 1
31-)etkileri --> 1
32-)isinmanin --> 1
33-)Kuresel --> 1

```

4.KARŞILAŞILAN SIKINTILAR ve ÇÖZÜMLERİ

1-)Sona Eklemede Eklediğim Elemanı 2.Kez Eklendiğinde Listede Kelimasayacini '+1' Artırması Lazımken Yeni Eleman Gibi Eklemesi :

Çözüm:

```

if (strcmp(temp->veri, kelime) == 0){

    kontrol=true;

    temp->kelimaSayaci = temp->kelimaSayaci+1;

    printf("Girdiginiz Kelimenin Aynisi Bagli
Listede olduğu için kelimeniz araya eklenemedi
sadece kelimenin sayaci arttırildi. !\n");

    return kelimeListe;

}

```

son eleman kontrolü yapılmadığı için bu hata çıkıyormuş(döngü dışına ekledik 1 tane daha bu kontrolden bu hatanın olmaması için).

2-)Bagli Listeyi Listelerken Tekrardan Sıraya Sokmayarak Ekliyordu Yeni Eklenen Kelimeleri

Çözüm:

```

while(kullanici_giris_menu != 5)

{

    currword3 = mykelime_t;

    .

    .

    .

    .

}

```

bu yüzden menü döngüsünün altında eski değerleri orda tanıttığımızda her yeni değişiklikle yeniden sıraya sokuyor.Yukarıdaki kod parçacığıyla bu sorunu çözdük.

5.SONUÇ

Bu proje sayesinde Bağlı Liste yapısının nasıl kurgulanıp nasıl yapısının oluşturulduğunu öğrendik.Bağlı listeye kelime ekleme yapılarıyla bağlı listeye erişimlerin nasıl kullanıldığını kavradık.

6.REFERANSLAR

1.Struct ve Typedef Yapısı ve İşlevleri:

<https://www.youtube.com/watch?v=AuZFMypOdvE>

<https://turkmuhendis.net/cprogramlama/yapi/>

2.Struct ve Stringler Yapısı ve İşlevleri:

<https://www.youtube.com/watch?v=QEkPQqYZryk>

3.Veri Yapılarına Giriş ve Bağlı Listeler ile İlgili Konu Anlatım:

<https://www.youtube.com/watch?v=r3uOBb3BM-0>

https://drive.google.com/file/d/11Z_N3uhVAm9wGn1IQDsd2lrWhvVBb1mF/view

4.Code Blokcs Hakkında Bilgi

<https://tr.wikipedia.org/wiki/Code::Blocks>

5.Akış Şeması Yapmak İçin Yardım

<https://www.tasarimkodlama.com/algorithma/flowchart-ornekleri/>

6.Akış Şeması Yapımı

<https://app.diagrams.net/>

7.Türkçe Arayüz

https://www.bilgigunlugum.net/prog/cprog/c_ornek/c_ornek00015

8.Bağlı Liste Yapısı Hakkında Genel Bilgi

https://notpast.com/c_programlama/Bagli-Listeler-72.html

7.AKIŞ ŞEMASI

