

**BAĞLI LİSTE PROJESİ**  
**Yasin Ömer KARA – Tahir Umut DOLAŞ**  
**Bilgisayar Mühendisliği Bölümü**  
**Kocaeli Üniversitesi**  
**180201077 - 180201053**

[y.omerkara1136@gmail.com](mailto:y.omerkara1136@gmail.com) – [tahirdolas41@gmail.com](mailto:tahirdolas41@gmail.com)

## ÖZET

Bu projede bizden C programlama dili kullanılarak bir bağlı liste uygulaması yapılması istendi. CodeBlocks kullanarak projemizi oluşturduk.

Projede istenen text içerisinde kelimeler okunacak ve kelimedeki text içinde kaç adet olduğu tutularak büyükten küçüğe doğru bağlı listeye eklenecekti.

Bağlı listede başa ekleme sona ekleme ve araya ekleme methodlarının kullanılması istendi.

Uygulamada her seferinde text içerisinde farklı bir kelime aratılacaktı.

Text içerisinde öncelikle kelime bulunup sonra kelimedeki kaç tane olduğu bulunup bu işlemin sonucunda text içerisine kelime eklenmeliydi.

**Bu projedeki amaç;** bağlı liste yapısını kullanmaktır. Başa, araya ve sona ekleme methodlarının kullanımı ile bağlı listede fonksiyon kullanımını ve yazımını gördük.

Ayrıca proje bağlı liste içerisinde kelime aratmayı ve dosyayı kelime kelime okuyarak bağlı listede tutmayı amaçlıyordu.

## 1.GİRİŞ

Öncelikle proje için bağlı liste kavramını araştırdık.

### **Bağlı Liste Nedir?**

Bağlı liste, her elemanın bir değerinin yanında bir de referans içerdiği veri yapısıdır. Bağlı listeler en basit ve en çok kullanılan veri yapılarından biridir. Yığın, kuyruk gibi diğer soyut veri yapılarını gerçeklemek için sıklıkla kullanılır.

Bağlı liste, düğüm (node) ismi verilen bir elemanda veri saklar. Yeni veri ekleneceğinde bir düğüm oluşturulur ve listeye bağlanır. Eleman silineceğinde ise düğümlerden birisi silinir, kopan bağ tekrar sağlanır.

Düğüm elemanlarına ulaşmak için liste üzerinde gezilir.

[Metni yazın]

Bizim projemizde düğüm(node) isim, sayaç ve next değerlerini struct yapısı ile tutuyordu.

## 2. YÖNTEM

Öncelikle dosya okuyarak projeye başladık. Dosyadaki kelimeleri kelime kelime okuttuk ve kelimedenden dosyada kaç tane olduğunu arattık. Daha sonra bu değerleri bulduktan sonra struct ile düğüm oluşturduk. (Şekil A.1)

Şekil A.1:

```
typedef struct dugum{
    int sayac;
    char isim[20];
    struct dugum* next;
};
```

Daha sonra bagliListe() fonksiyonu ile kelimeleri birbirine bağladık. Bu fonksiyon içerisinde başa, araya ve sona ekleme işlemlerini içeriyor. (Şekil A.2)

Şekil A.2:

```
///Baqli Liste ile kelimeleri baglama fonksiyonu
void bagliListe(struct dugum** ilk, struct dugum*kelime)
{
    struct dugum*simdiki;

    if (*ilk == NULL)
        *ilk = kelime;

    else if ((*ilk)->sayac >= kelime->sayac) {
        kelime->next = *ilk;

        *ilk = kelime;
    }

    else {
        simdiki = *ilk;

        while (simdiki->next != NULL && simdiki->next->sayac < kelime->sayac)
            simdiki = simdiki->next;

        kelime->next = simdiki->next;
        simdiki->next = kelime;
    }
}
```

Bu fonksiyonu kullanırken aynı zamanda bellekten düğümden yer almamız gerekiyordu. Bunun için dugum\_al() fonksiyonu yazdık.(Şekil A.4)

Şekil A.4:

```
struct dugum* dugum_al(int sayac,char isim[20])
{
    struct dugum* kelime = (struct dugum*)malloc(sizeof(struct dugum));

    kelime->sayac = sayac;
    strcpy(kelime->isim, isim);
    kelime->next = NULL;
    return kelime;
}
```

[Metni yazın]

Daha sonra kelimeleri dosya içerisinde aramak için kullanıcıdan kelime aldık. Ve ara() fonksiyonu ile arama işlemini gerçekleştirdik.

(Şekil A.4)

Şekil A.4:

```
///KELİME ARAMA FONKSİYONU
struct dugum* ara(struct dugum**birinci, char isim[]){
    struct dugum* aranan = NULL;
    struct dugum* kopyala= *birinci;

    while(kopyala!=NULL){
        if(strcmp(kopyala->isim, isim)==0){
            aranan = kopyala;
            return aranan;
        }
        kopyala=kopyala->next;
    }
    return aranan;
}
```

### 3. SONUÇ

Bu projede dosyadan kelime okumayı, kelime sayısını saydırmayı öğrendik.

Bağlı liste yapısı kullanılarak bu kelimeler birbirlerine nasıl bağlanırdı gördük.

Bağlı listede düğüm(node) nasıl oluşturulur kelimeler nasıl eklenirdi öğrendik.

Aynı zamanda bağlı liste içerisinde arama nasıl yapıldı öğrendik.

Aynı zamanda başa, araya ve sona ekleme sayesinde kelimeleri nasıl sıralayacağımızı öğrendik.

### 4. DENEYSEL SONUÇLAR

Aramak istediğiniz kelimeyi girin:

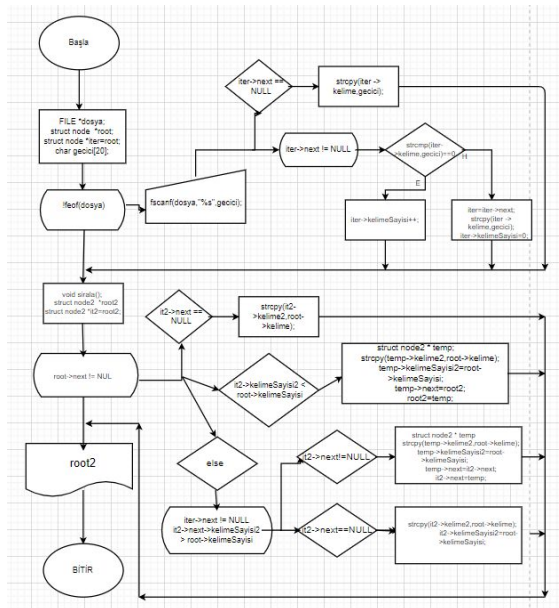
```
En
Kelime bulundu....
Kelime: En --> Adet: 3
1. * --> 5
2. ntv.com.trâ?tdede --> 1
3. birazdan --> 1
4. oldu --> 1
5. rekor --> 1
6. ayında --> 1
7. yılının --> 1
8. 2020 --> 1
9. ardından --> 1
10. 2019'un --> 1
11. görülüyor --> 1
12. net --> 1
13. yıl --> 1
```

Projede sıralama konusunda başlangıçta sorunlar yaşadık. Başa, sona ve araya ekleme methodlarımızı güçlendirerek bu sorunu daha sonra çözdük.

Projede her bir düğüm için yer alma konusunda başlangıçta sıkıntı yaşasak da detaylı araştırmalar sonucunda bu işlem için bir fonksiyon yazarak bu problemi hallettik.

[Metni yazın]

## 5. AKIŞ ŞEMASI



## 6. KULLANDIĞIMIZ KÜTÜPHANELER

- `#include <stdio.h>`
- `#include <stdlib.h>`
- `#include <locale.h>`
- `#include <ctype.h>`

## 7. KAYNAKÇA

[1]. Web Site

<http://bilgisayarkavramlari.com/2007/05/03/linked-list-linkli-liste-veya-bagli-liste/>

[2]. Web Site

[https://notpast.com/c\\_programlam/a/Bagli-Listeler-72.html](https://notpast.com/c_programlam/a/Bagli-Listeler-72.html)

[3]. Web Site

[https://www.bilgigunlugum.net/prog/cprog/c\\_dosya](https://www.bilgigunlugum.net/prog/cprog/c_dosya)

[4]. Web Site

<https://nerdbook.wordpress.com/2018/03/21/bagli-listeler/>