

# Kocaeli Üniversitesi

## Bilgisayar Mühendisliği Bölümü

### Programlama Laboratuvarı I

## Counting Words with Linked Lists

*Berk Sunduri*

*[180201145@kocaeli.edu.tr](mailto:180201145@kocaeli.edu.tr)  
[berksunduri@gmail.com](mailto:berksunduri@gmail.com)*

### Projenin Özeti:

Programlama laboratuvarı I olarak bizden bağlı listelerle kelime saymamız istenmektedir.

Ben proje için C programlama dilini ve Code::Blocks geliştirme ortamını seçtim.

Proje dokümanında bizden beklenen beş ayrı istek var. Bunlar: uygulamada metin içerisinde geçen kelimeler sayılacak, uygulamada tekrarlanan kelime bulunacak, adet sayısı hesaplanacak ve büyükten küçüğe doğru listeye eklenecek. Bağlı liste kelime ve adet bilgisi içermeli, büyükten küçüğe doğru sıralanmalı ve bağlı listede başa, sona ve araya ekleme metotlarının tümünü yazmamız gereklidir.

Projede en zorlayıcı isteri ise diziler kullanımının yasak olmasıydı.

### 1. GİRİŞ

Proje için C programlama dili ve Code::Blocks geliştirme ortamını kullandım.

C programlama dili; genel amaçlı, sözcüksel değişken kapsamını ve özyinelemeyi destekleyen bir programlama dilidir.

Code::Blocks platformu; içerisinde C,C++ ve Fortran yazabileceğiniz ücretsiz bir entegre geliştirme ortamıdır(IDE). Code::Blocks esnek bir IDE'dir ve kişiye özel bir sürü konfigürasyon sunar.

### 2. TEMEL BİLGİLER

Projemde;Nodeları kaydetmek için kullandığım 2 tane Struct ve Main fonksiyonunu saymazsak tam 14 fonksiyon var. Bunlar: AppendListFromFile(), RewindFile(), RewindFile(), OpenFileRead(), display(), delete(), add(), append(), disp\_occur(),occur(), count(), swap(), swap2() ve bubbleSort(), insertNodeAtPosition().

#### Struct node:

Oluşturduğum struct yapısı bağlı listeyi kaydetmemde işe yarıyor.

#### Struct node\_occur:

Oluşturduğum struct yapısı diğer struct yapısından farklı olarak kelimelerin kaç defa tekrarlandığı bilgisinide içeriyor.

#### AppendListFromFile()

Bu fonksiyon okunacak dosya içinde ki kelimeleri bağlı listeye aktarır.

#### RewindFile()

Bu fonksiyon dosyayı başa sarmamızda işimize yarar.

### **OpenFileRead()**

Bu fonksiyon okumak için istediğimiz dosyayı açar..

### **display(struct node)**

Yolladığımız bağlı listeyi terminale yazdıran fonksiyon.

### **delete(char num[])**

Bu fonksiyon bağlı listede istediğimiz bir node'u silmemizde işimize yarar.

### **add(char num[])**

Bu fonksiyon oluşturulan listenin başına node eklememizde işimize yarar.

### **append(char num[])**

Bu fonksiyon oluşturulan listenin sonuna node eklememizde işimize yarar.

### **disp\_occur(struct node\_occur)**

Bu fonksiyon oluşturulan bağlı listede bulunan tekrarlayan kelimeleri terminale yazdırmakta kullanılır.

### **occur(struct node, struct node occur)**

Bu fonksiyon oluşturulan bağlı listede tekrarlayan kelimeleri bulmamızda işe yarar. Bu kelimeleri ana bağlı listeden farklı bir bağlı listeye sayım için kopyalar.

### **count()**

Bu fonksiyon bağlı listede kaç node olduğunu öğrenmemizde işimize yarar.

### **swap2(data)**

Bu fonksiyon listeyi sıraladığımızda node datalarının sıralanmasını sağlar.

### **swap(counter)**

Bu fonksiyon listede kelimeleri tekrar sayısına göre sıralamamızda işe yarar.

### **BubbleSort(struct node\_occur)**

Bu fonksiyon kopyalanan diğer listeyi sıralamamızda işe yarar

### **insertNodeAtPosition()**

Bu fonksiyon girilen verinin istenilen herhangi bir pozisyona eklenmesinde yardımcı olur.

## **3. YAPILAN ARAŞTIRMALAR**

Projeye başladığımda karşılaştığım en büyük sorun dizi kullanamamaktı. Bu kısıtlamanın altından kalkmak için bağlı listeleri sadece C programlama dilinde değil birkaç programlama dilinde araştırmak oldu. Bulduğum çözüm 2 tane bağlı liste kullanmak oldu. Yaptığım araştırmalar sonucunda kendimi bağlı listeler konusunda epeyce geliştirdim.

```
struct node
{
    char data[MAX];
    struct node *next;
};

|
struct node_occur
{
    char data[MAX];
    int times;
    struct node_occur *next;
} node_occur;
```

Karşılaştığım bir diğer sorun ise listeyi doğru sıralamaktı. Bunu ilk önce Insertion Sort ile yapmayı denedim fakat Insertion Sort'u nodelere implement etmekte başarısız oldum. Bende daha kolay olan Bubble Sort alternatifine yöneldim. Zorlandığım yerlerde internetten yardım

almak zorunda kaldım.(İnternetten yardım aldığım kısımlar kodda belirtildi.)

```
void bubbleSort(struct node_occur *start)
{
    int swapped, i;
    struct node_occur *ptr1;
    struct node_occur *lptr = NULL;

    if (start == NULL)
        return;

    do
    {
        swapped = 0;
        ptr1 = start;

        while (ptr1->next != lptr)
        {
            if (ptr1->times < ptr1->next->times)
            {
                swap2(ptr1, ptr1->next);
                swap(ptr1, ptr1->next);
            }
        }
    }
}
```

### Kaba Kod:

1. Menüden İşlem Seç.
2. Case1: Listenin sonuna node eklemek.
3. Case2: Listenin başına node eklemek.
4. Case3: Liste elemanlarını görüntüle.
5. Case4: Listenin boyutunu göster.
6. Case5: Listedeki kelimeleri silmek istediğiniz kelimeyi yazın.
7. Case6: Programı durdur.
8. Case7: Okumak istediğin dosyayı aç.
9. Case8: Dosyada ki kelimeleri listeye aktar.
10. Case9: Dosyada farklı işlem yapmak için dosyayı geri sar.
11. Case10: Listedeki tekrarlanan kelimeleri ve tekrar sayısını göster.
12. Case11: Dosyayı kapat.
13. Case12: Listenin istenilen noktasına node eklemek.

**Kod akış şeması ekteidir.**

### 5.Genel Yapı:

Program başlatıldığında karşınıza altta gösterilen menü çıkacaktır.

```
*****Liste İşlemleri*****
1.Sona ekle      2.Basa Ekle      3.Liste Görüntüle
4.Liste Boyutu   5.Sil      6.Cıkıs
7.Dosya Oku      8.Dosyadan Listeye Aktar  9.Dosyayı Geri Sar
10.Tekrarlayan Kelimeler 11.Dosyayı Kapat 12.Ortaya Ekle

Yapmak istediğiniz işlemi seçin : (0: Menu)
```

Bu menü sayesinde yapmak istediğiniz işlemi kolayca seçebilirsiniz. İlk başta dosya okumayı seçerek program dosyasına eklediğimiz herhangi bir yazıyı programa okutuyoruz. Daha sonra Dosyadan listeye Aktarı seçerek dosya içinde ki kelimeleri listeye aktarıyoruz. Bundan sonrada projenin amacı olan Tekrarlayan Kelimeleri seçiyoruz program otomatik olarak bize kelimeleri sayıyor ve büyükten küçüğe doğru sıralıyor. En sonda programı kapatıyoruz.

### 6.Kaynakça:

#### Kitap:

A'dan Z'ye C Kılavuzu - PUSULA  
YAYINCILIK VE İLETİŞİM - Kaan Aslan

C How to Program 6<sup>th</sup> Edition- Paul Deitel and Harvey Deitel

#### Website:

StackOverFlow- Insert a node at specific position in a linked list in C (only)  
(<https://stackoverflow.com/questions/62063574/insert-a-node-at-specific-position-in-a-linked-list-in-c-only>)

GeeksForGeeks-Linked List Data Structure  
(<https://www.geeksforgeeks.org/data-structures/linked-list/>)

Carnegie Mellon University-Linked Lists  
(<https://www.cs.cmu.edu/~guna/15-123S11/Lectures/Lecture09.pdf>)

Hawaii University-Linked Lists(<https://www2.hawaii.edu/~walbritt/ics212/examples/linkedlist.c>)

DaniWeb- How to Read Linked Lists from a Text File.  
(<https://www.daniweb.com/programming/software-development/threads/471753/reading-from-txt-file-into-linked-list>)

XSPDF-Find Duplicates in C

(<https://www.xspdf.com/resolution/59594570.html> )

Wikipedia-Linked Lists

([https://en.wikipedia.org/wiki/Linked\\_list](https://en.wikipedia.org/wiki/Linked_list) )