

**KOCAELİ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**PROGRAMLAMA LABORATUVARI-1**

**1.PROJE**

**Muhammed Koç**

190202040

**Osman Şafak**

190202064

**Proje Özeti:**

Projede bizden txt olarak verilen bir belgeyi okuyarak belgede geçen metnin kelimelerini belirleyip bir bağlı listeye atayarak ,aynı zamanda da o kelimenin cümlede kaç defa geçtiğini sayıp bağlı listeye ekleyen bir program yapılması istenmiştir.

Kelimeler bağlı listede tekrar sayıları büyükten küçüğe doğru sıralı olacak şekilde dizilecektir. Programda 3 adet fonksiyon bulunacaktır. Fonksiyonlar başa ekleme ortaya ekleme sona ekleme görevlerini yapacaklardır. Son olarak bu liste başka bir txt dosyasına basılacaktır. Türkçe karakter ve büyük küçük harf önemsenmeyecektir.

Proje C dili üzerinden yapılacaktır.

Çok boyutlu [][] dizi kullanılmayacaktır.

Ama Char [] dizisi kullanılabilir.

Proje 2 kişilik grup halinde yapılacaktır.

**GİRİŞ**

Projede struct ve tek yönlü bağlı liste kullanıldı.

Projede C dili kullanıldı.

Çok boyutlu dizi kullanılmadı.

Struct içinde bir char dizisi ve birde integer değer tutuldu.

Projede yazılan txt dosyasında Türkçe karakter sorunu yaşanmadı.

Codeblocks ide si kullanıldı.

## Yöntem

Projede 1 struct, 1 tek yönlü bağıl, liste 4 ara fonksiyon kullanılmıştır.

Göster fonksiyonu bağıl listeyi ara yüzde gösterir.

Başa ekle fonksiyonu gönderilen kelimeyi ve tekrar sayısını listenin en başına ekler.

Sona ekle fonksiyonu gönderilen kelimeyi ve tekrar sayısını bağıl listenin en sonuna ekler.

Ortaya ekle fonksiyonu gönderilen kelimeyi ve tekrar sayısını en uygun yere (büyükten küçüğe sıralamaya göre) ekler.

Eklemeler fonksiyonu bağıl liste boşsa ilk elemana gönderilen kelimeyi ve tekrar sayısını atar ana fonksiyondan gönderilen her kelime ve tekrar sayısı burada güncellenir. Diğer 3 fonksiyon eklemeler fonksiyonundan yönlendirilir.

Main fonksiyonunda kaç kelime olduğu bulunur. Txt dosyasından okunan kelimeler eklemeler fonksiyonuna gönderilir. Ve oradan uygun fonksiyona yönlendirilerek bağıl listeye kaydedilir.

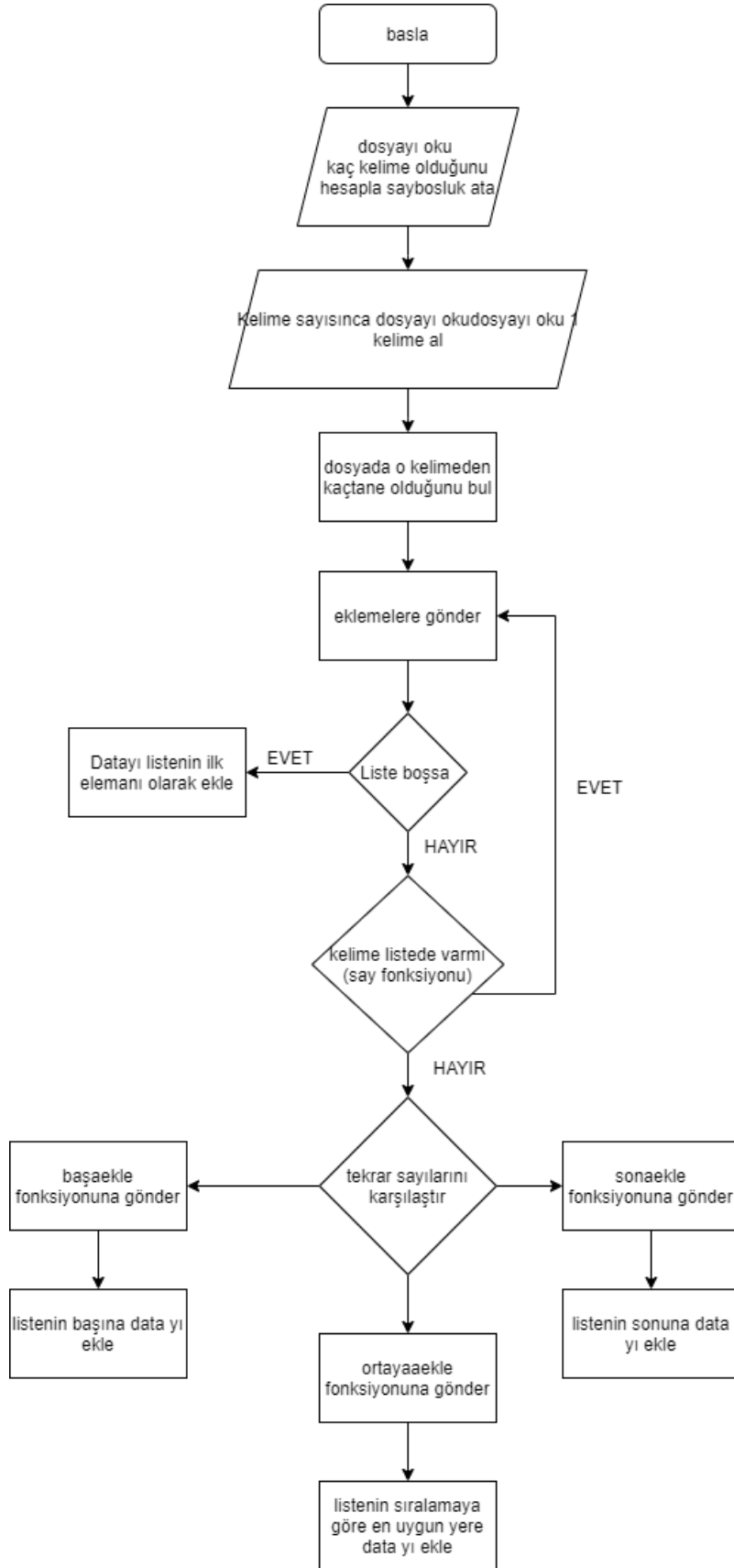
Eklemeler fonksiyonun yönetilmesi ile bağıl listedeki veriler büyükten küçüğe sıralanmıştır.

Struct içinde pointer kullanılarak malloc fonksiyonu ile string e yer açılmış ama sonradan pointer güncellenmesi sonucu, onu tutan değişkenlerinde sürekli değişmesiyle, bağıl listede hep en son yapılan değişiklik kaydedilmiş ,eski kelimeler kaydedilemediği için tek boyutlu dizi kullanımına geri dönmüştür.

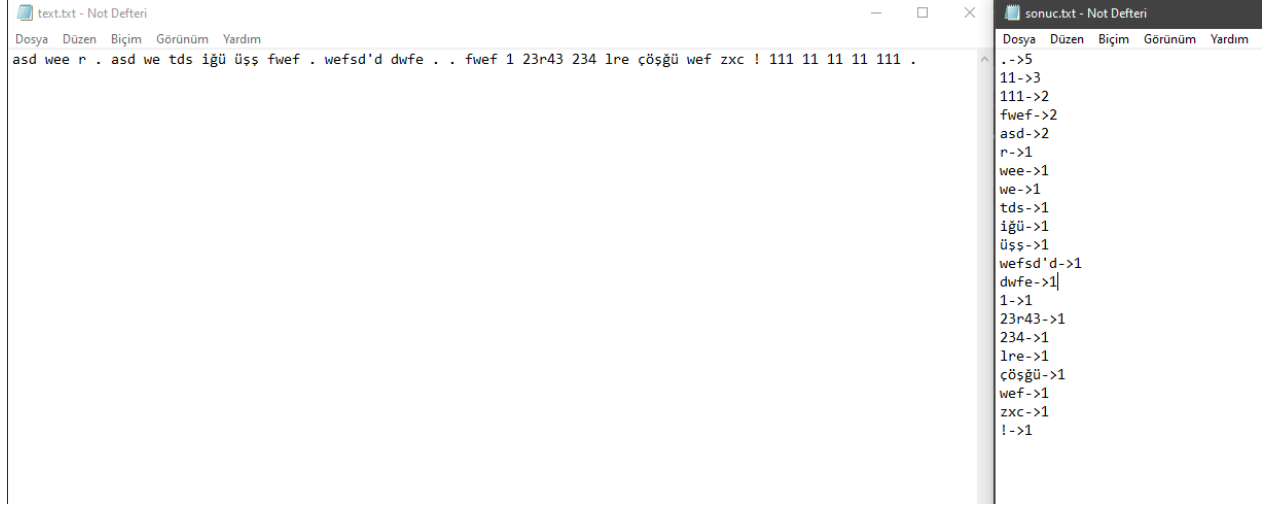
Projede ilk başta fscanf fonksiyonunun “\n” harfiyle diğer kelimeleri seçebildiği düşünüldüğü için önce metni okunup düzenlenerek, tüm kelimelerin sonuna “\n” eklenerek tekrar yazılmış ve düzenlenmiş dosyadan tekrar okutulmuştur. Ama sonradan fscanf fonksiyonun boşluk karakterine de duyarlı olduğu fark edilerek koddan yapılan gereksiz işlem çıkarılmıştır.

Txt dosyasında metnin sonuna boşluk bırakılmamalıdır.

## Akış Şeması

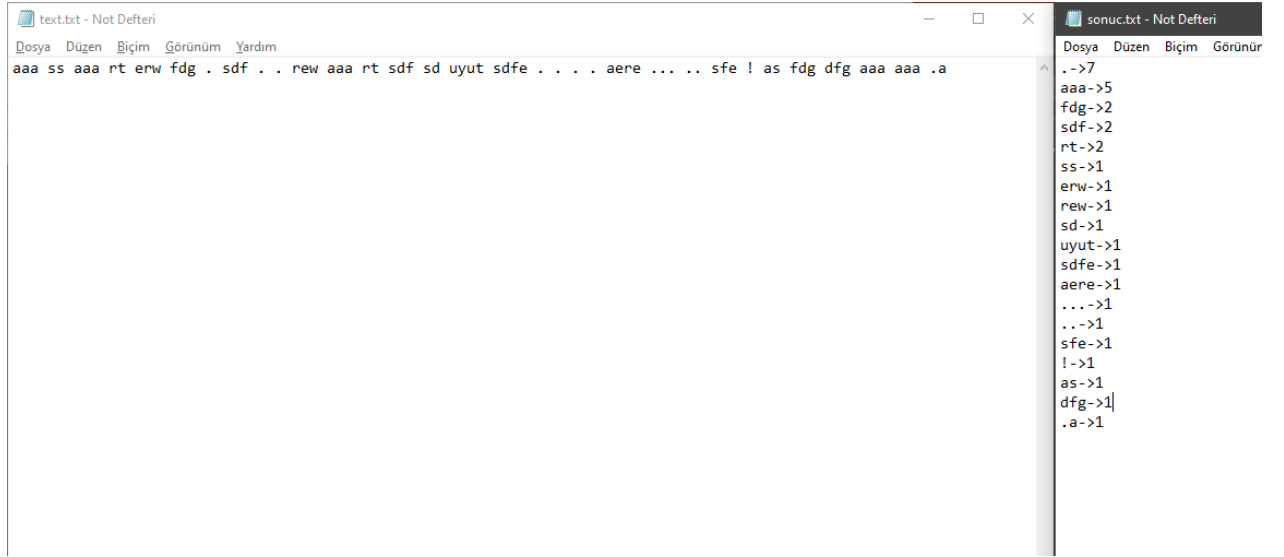


## Deneysel Sonuçlar



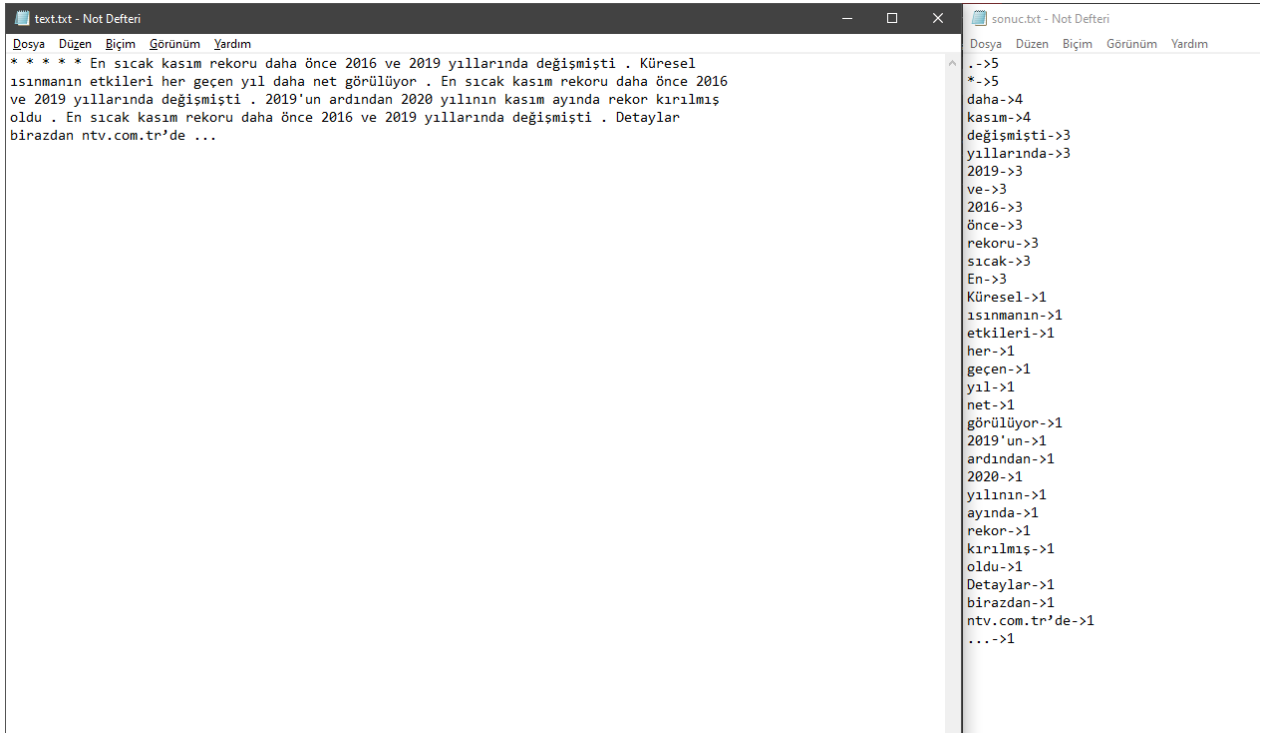
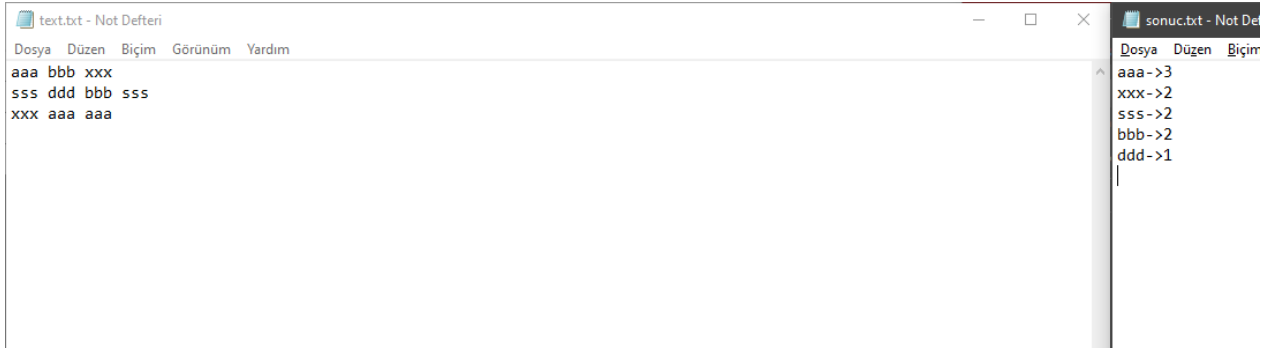
The image shows two Notepad windows side-by-side. The left window, titled 'text.txt - Not Defteri', contains the following text: 'asd wee r . asd we tds iğü üşş fwef . wefsd'd dwfe . . fwef 1 23r43 234 lre çöşğü wef zxc ! 111 11 11 11 111 .'. The right window, titled 'sonuc.txt - Not Defteri', displays the frequency analysis of the text in the left window. The analysis is as follows:

Character	Frequency
.	5
11	3
111	2
fwef	2
asd	2
r	1
wee	1
we	1
tds	1
iğü	1
üşş	1
wefsd'd	1
dwfe	1
1	1
23r43	1
234	1
lre	1
çöşğü	1
wef	1
zxc	1
!	1



The image shows two Notepad windows side-by-side. The left window, titled 'text.txt - Not Defteri', contains the following text: 'aaa ss aaa rt erw fdg . sdf . . rew aaa rt sdf sd uyut sdfe . . . . aere ... .. sfe ! as fdg dfg aaa aaa .a'. The right window, titled 'sonuc.txt - Not Defteri', displays the frequency analysis of the text in the left window. The analysis is as follows:

Character	Frequency
.	7
aaa	5
fdg	2
sdf	2
rt	2
ss	1
erw	1
rew	1
sd	1
uyut	1
sdfe	1
aere	1
...	1
..	1
sfe	1
!	1
as	1
dfg	1
.a	1



## SONUC:

Bağlı listenin yapısı pekiştirilmiş. Pointer kullanımı ve pointer ile bellekte yer açma işlemleri anlaşılmış dizi mantığı kavranmıştır.

Bağlı listelerde karşılaşılabilecek olası hatalar tecrübe edilmiştir.

Grup çalışması hakkında ilk tecrübe edinilmiş grup halinde proje yapmada karşılaşılabilecek sorunlar ve getirileri deneyimlenmiştir.

Dosyadan metin okuma ve dosyaya yazdırma işlemleri pekiştirilmiştir.

## KAYNAKÇA

### web

[C Veri Yapıları: Generic Bağlı Liste | by Sıddık Açıl | Medium](#)

[https://notpast.com/c\\_programlama/Bagli-Listeler-72.html](https://notpast.com/c_programlama/Bagli-Listeler-72.html)

[Linked List \(Linkli Liste veya Bağlı Liste\) – Bilgisayar Kavramları \(bilgisayarkavramlari.com\)](#)

[strcmp Fonksiyonu | C ve Sistem Programcıları Derneği \(csystem.org\)](#)

[2. Hafta - C # \(weebly.com\)](#)

[xxx.drawio - diagrams.net](#)