

KOCAELİ ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ



PROGRAMLAMA LABORATUVARI I

PROJE 1

-ÖĞRENCİ BİLGİ SİSTEMİ PROGRAMI-

200202017 Mehmet Ali AKDOĞAN

200202098 Muhammed Numan MERCAN

## Özet

Programlama Laboratuvarı 1 dersinin ilk Projesi C dilinde dosya işlemlerinin daha iyi kavranması için bir okulda öğrencilere ait bilgileri veri dosyası olarak binary , index dosyası olarak text şeklinde tutulan ve bu veriler ile işlem yapabilmeyi amaçlayan (kayıt ekleme , silme , güncelleme vb) bir projedir. Bu proje ile farklı dosya yapıları üstünde işlem yapmayı , verileri sıralamayı , struct tanımını uygulamış ve öğrenmiş olmayı amaçlıyoruz.

Bu rapor ile de projeyi yaparken hangi süreçlerden geçtiğimizi hangi metodları kullandığımızı açıklamayı ve projeyi anlaşılır kılmayı hedefledik.

## GİRİŞ

Projede istenen , bir struct ile öğrenci kayıt ve veri tutma çalışmamızda bu verilere yenisini ekleme,istenilen veriyi kontrol etmek,silmek veya güncellemek gibi bazı metodlardın.

Tabi bu verileri kayıt ederken bir binary(.bin,.dat,.doc) dosyaya kaydedilmesi ve ardından bir indeks(txt) dosyasına bu verileri sıralı bir halde yazdırılıp istenen işlemleri bu dosya üzerinden yapılması bekleniyor.

**kayıtEkle** : veri dosyamına yani binary file ' a yeni öğrencinin bilgileri eklenir.

**kayıtBul** : sıralı olan indeks dosyasından binary search algoritması ile aranan kayıt bulunur.

**kayıtSil** : verilen anahtardaki bilgilere sahip öğrenci kayıttan silinecek.

**kayıtGuncelle** : Numarasi belirtien öğrencinin yalnızca puanı güncellenebilecek.

**veriDosyasiniGoster** : .bin uzantılı dosyadaki tüm kayıtları listeler.

**indeksDosyasiniGoster** : .txt uzantılı dosyadaki (sıralı) tüm kayıtları gösterir

**indeksDosyasiSil** : .txt uzantılı dosyayı siler.

Bu isterleri hepsini ayrı birer fonksiyonda tanımladık , menü yardımı ile seçilmek üzere hazırladık.

```
1 - Indeks Dosyasi Olustur
2 - Kayit Ekle
3 - Kayit Bul
4 - Kayit Sil
5 - Kayit Guncelle
6 - Veri Dosyasini Goster
7 - Indeks Dosyasini Goster
8 - Indeks Dosyasi Sil
0 - Cikis
-----
Yapmak istediginiz islemi secin : _
```

## İlerleyiş-Yöntem

### - Başlamadan Önce

Projeye ve kodlamaya iyice girmeden önce metodları oluşturmak için olası yöntemler araştırıldı . Adresleme metodları tekrardan gözden geçirildi. Dosya metodları denendi.Temel alıştırmalar ve araştırmalar yapıldı.

### - Başlangıç ve İlerleyiş

İstenen metodlar , dosya yapıları , veri girdileri , algoritmalar oluşturuldu. Switch yapısıyla her fonksyon için temiz bir komut ekranı sağlandı.Fonksiyonları zaten alışık olduğumuz txt dosya işlemleriyle biraz denedikten sonra binary dosya yapısına geçirdik ve düzenli çalıştırmayı başardık.

Bu aşamada birkaç fonksiyonumuzuda çalışır şekilde entegre edebildik.

Kayıt sil , kayıt güncelle ve kayıt ekle çalışır hale geldiler.Binary dosya yapısından txt ye çevirerek indeks.txt dosyamızı oluşturduk ve bu şekilde kayıtBul fonksiyonu için kullanacağımız sıralı dosyaya da sahip olduk.

```
vPtr = fopen("veri.bin","a+b");

fwrite(&ogr,sizeof(struct veri),1,vPtr);

fread(&ogr,sizeof(struct veri),1,vPtr)
```

Sekil-1 (.bin uzantılı işlemler)

Bu aşamalarda oluşturacağımız algoritmalar için yalancı kod ve onunla birlikte akış şeması oluşturmaya başladık biraz vakit alsada projede daha anlaşılır ve hızlı gitmemizi sağlayan bu metodları raporun ilerleyen kısımlarında paylaşıyor olacağız.

İlk olarak .bin Veri dosyasına kayıtlarımız en sondan eklemeli şekilde kaydediliyor.

Ogr NO	Adres	Ders No	Puan
200202098	b26db0	101	100
181715023	b214c0	109	99
200202001	b214e0	104	84
200202058	1d14f0	105	46

Sekil-2 (Veri File)

İndeks dosyasına veriler önce sıralama algoritması ile numaralara göre sıralanıp o şekilde listeleniyor

Ogr No	Adresi	Ders Kodu	Puan
181715023	b214c0	109	99
200202001	b214e0	104	84
200202058	1d14f0	105	46
200202098	b26db0	101	100

Sekil-3 (İndeks File)

Ogrencinin puanını kayıt güncelle fonksiyonu ile güncelliyoruz.

Bilgisi güncellenecek ogr no : 200202098			
Puanı : 85			
Ogrenci bilgileri degistirildi...			
200202098	b26db0	101	100
200202098	b26db0	101	85

Sekil-4 (kayitGuncelle());

Ogr No	Adresi	Ders Kodu	Puan
181715023	b214c0	109	99
200202001	b214e0	104	84
200202058	1d14f0	105	46
200202098	b26db0	101	85

Sekil-6 (.bin-.txt dosyaları içeriği)

```
struct veri
{
    int ogrNo;
    int *ogrNoAdres;
    int dersKodu;
    int puan;
};
```

Sekil – 7 (Struct Yapısı – düzenlenmemiş hal)

### -Karşılaşılan Problemler

Proje esnasında en yorulduğumuz yer sıralama algoritmasını txt de nasıl kullanacağımızdı. Çözüm için sabit dizi , çok boyutlu dizi, dinamik dizi , bağlı liste gibi pek çok yol denedik fakat sonunda satır sayısında tekabül eden elemanda 2 boyutlu dizi ile sıralamamızı hallettik. Bu sorunun ardından kayıtBul fonksiyonu bizi zorlayan kısım oldu , kayıtBul fonksiyonu ikili arama algoritması ile yapılması gerekiyordu.Proje sürecinde pek çok yol denendi sonuçlar raporun ilerleyen kısımlarında karşımıza çıkacaktır.

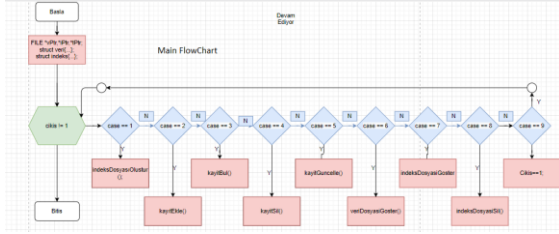
### -Akış Şeması

Akış şeması haliyle koda göre daha az ayrıntı içeriyor , Proje sürecinde kodlarımızda bir çok deneme ve değişikliğe başvurduğumuz içinde bazı değişiklikler akış şemasında gözden kaçmış olabilir. Fakat amacımız zaten algoritmanın en yalın haliyle mantığını görsel olarak ortaya koymak olduğu için şema bize yeterli oldu.Belirttiğimiz gibi akış şemasını ilerleyen sayfalarda detaylı bir şekilde raporumuzda bulundurduk.

### -Sonuç

Uygulanan onca denemeden ve hatadan sonra ortaya isterlerin neredeyse hepsini karşılayan ve düzgün çalışan bir proje çıktı. Bu süreçte özellikle dosya işlemleri , bunlara entegre sıralama ve yapıları iyice kavradığımızı , aldığımız onca hatanın pek çok şeyi öğrenmemizi sağladığını görmüş olduk.

Akış şeması fazla büyük olacağı için tek bir şema ile değil Ana şema ve onun alt şemaları olan fonksiyonların ayrı ayrı şemalarını oluşturmayı tercih ettik , aşağıda küçük örnekleri bulunuyor.



Sekil-8 (Main Şema)

Alt şemalardan birisini örnek olarak burda paylaşıyoruz tüm şemayı ilerleyen kısımlarda bir bütün olarak bulabilirsiniz.

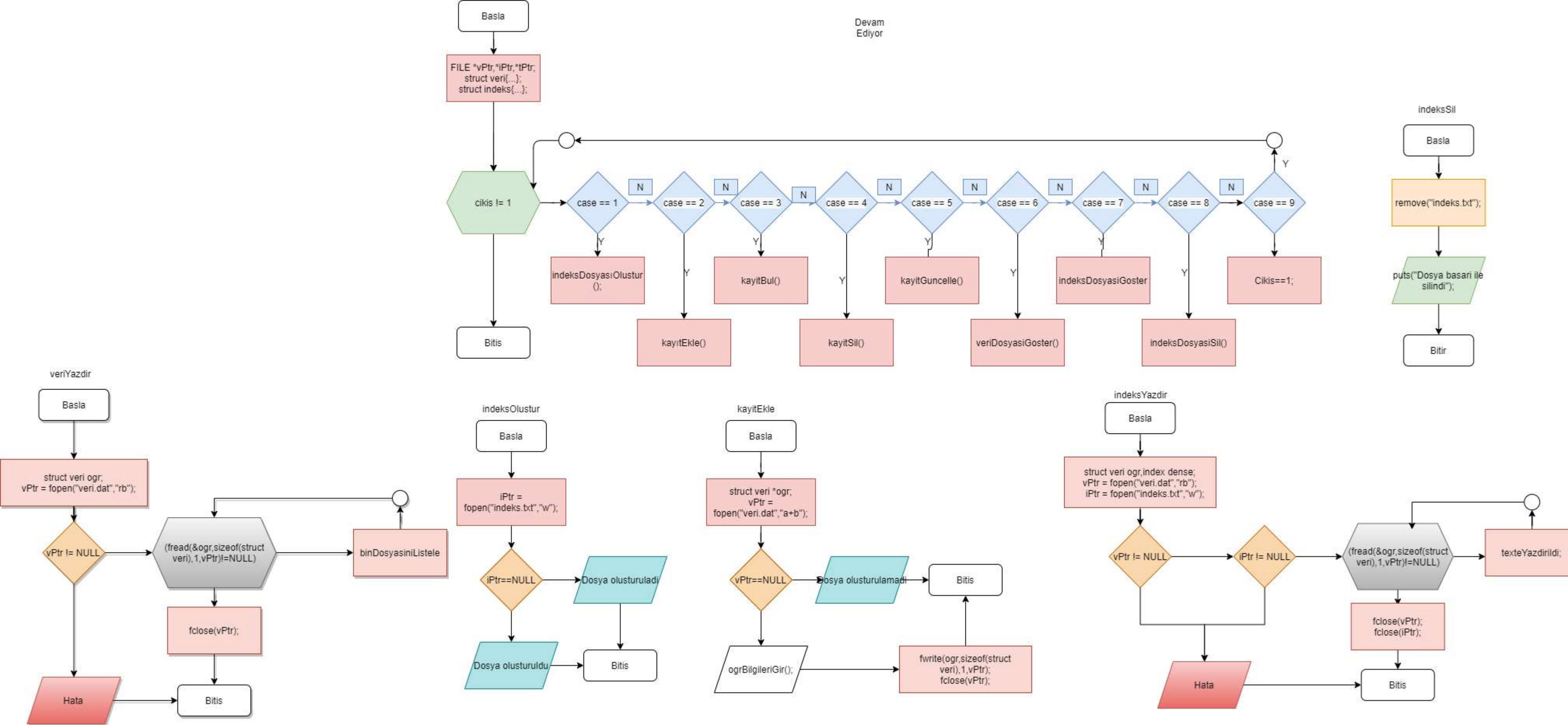


Çok detaylı olmasada projeye daha hiç başlamadan evvel akılda yalın bir fikir oluşması açısından fazlasıyla yüzeysel bir plan taslağı (çok basit pseudo kod örneği) oluşturduk.

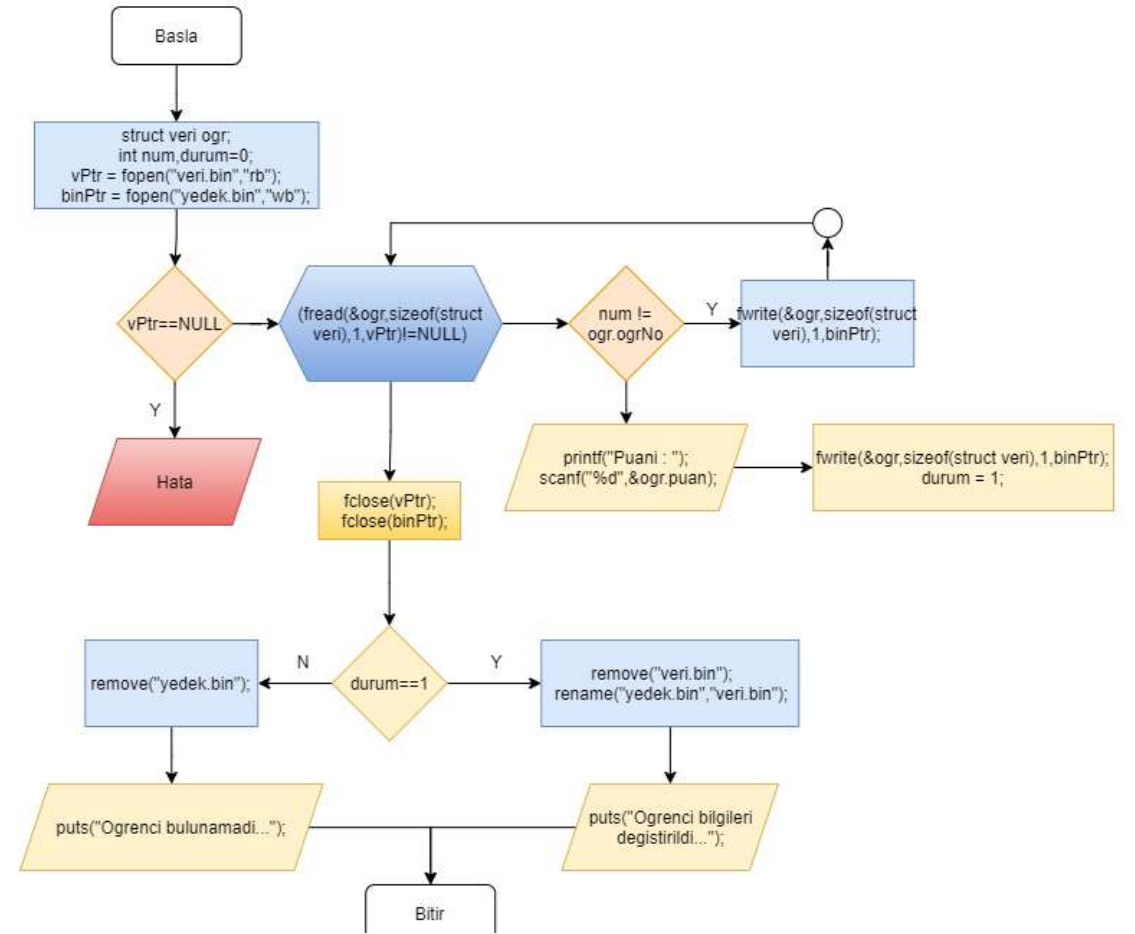
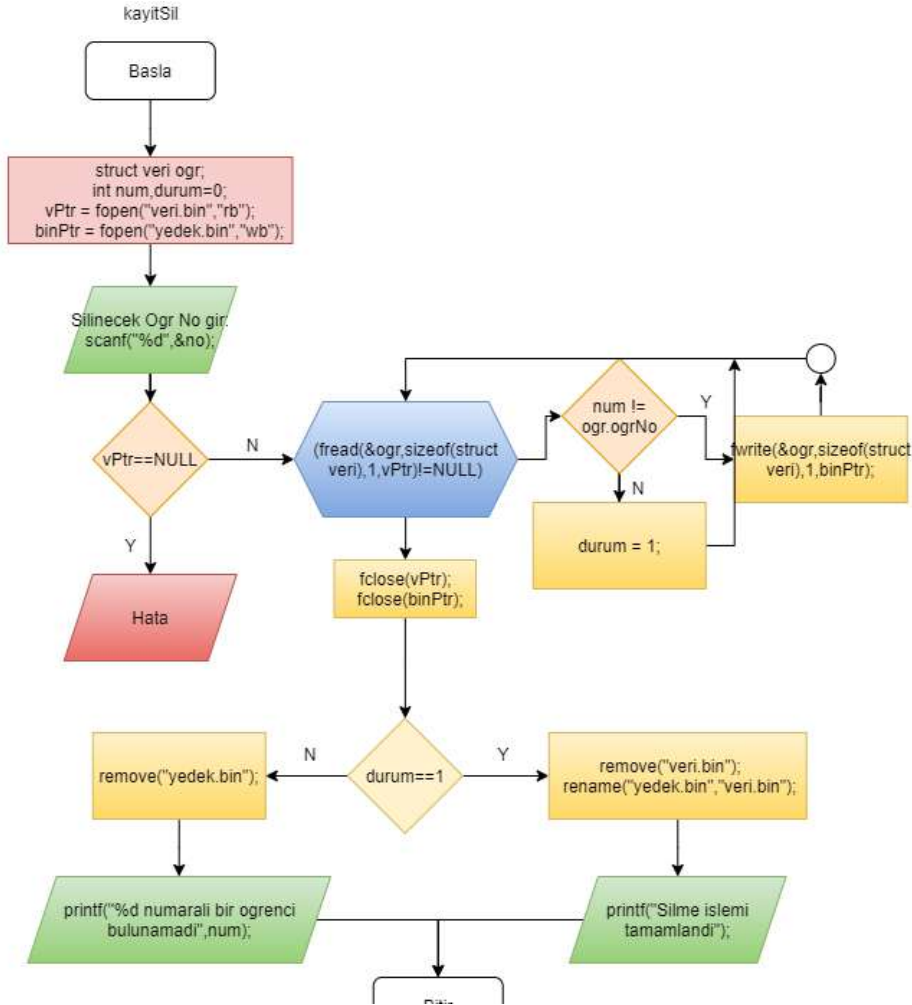
- 1- while döngüsünde programı aç,
- 2 - switch case yapısı ile fonksiyonları bölümlendir
- 3 - seçilen duruma göre içine girilen fonksiyonların şeması
- 3.0 - indeksDosyasıOluştur -> veri dosyasındaki kayıtlar için bir indeks dosyası oluşturulacak,
- 3.1 - kayıtEkle -> binary(veri) dosyasını aç sonuna kaydet , indeks (text) dosyasını güncelle
- 3.2 - kayıtBul -> istenilen anahtar veri dosyasında aranacak,
- 3.2.1 - İndeks dosyasında binary search ile arama
- 3.3 - kayıtSil -> Verilen anahtar(id) için ilgili doğru kayıt bulunup (kayıtBul gibi) silinecek.
- 3.4 - kayıtGüncelle -> Bir anahtar(id) için aranılan kayıt bulunup sadece puanı güncellenecektir
- 3.5 - veriDosyasıGöster -> binary(veri) dosyasını listeler.
- 3.6 - indeksDosyaGöster -> indeks(text) dosyasını listeler.
- 3.7 - indeksDosyaSil -> indeks(text) dosyasını siler

Sekil – 9 (Plan , başlangıç yalancı kodu)

Devam  
Ediyor







## KAYNAKÇA

- <https://web.cs.hacettepe.edu.tr/~maydos/Docs/c/dosyalar.pdf>  
Hacettepe Üni. Dosya İşlemleri
  - <https://web.itu.edu.tr/~coskun/contents/lessons/bil104/12.pdf>  
İstanbul Teknik Üni. Dosya İşlemleri
  - [https://www.youtube.com/c/Sadievrenseker\\_BK](https://www.youtube.com/c/Sadievrenseker_BK)  
[https://www.youtube.com/watch?v=2QmXifFCcGk&ab\\_channel=BilgisayarKavramlari](https://www.youtube.com/watch?v=2QmXifFCcGk&ab_channel=BilgisayarKavramlari) -> Bilgisayar Kavramları / Sadi Evren Seker
  - <https://www.youtube.com/channel/UCZvtPTYv5rwVI9qHpd6rnQ>  
Mucit Yazılım
  - <https://stackoverflow.com/>
  - <https://www.geeksforgeeks.org/>
  - <https://bilgisayarkavramlari.com/2008/08/09/siralama-algoritmaları-sorting-algorithms/>
  - [https://www.bilgigunlugum.net/prog/cprog/c\\_dosya](https://www.bilgigunlugum.net/prog/cprog/c_dosya)
  - <https://gurelahmet.com/c-dilinde-siralama-algoritmaları/>
- <http://cprogramarsivi.blogspot.com/>