# PROGRAMLAMA LABORATUVARI 1
# Proje 1

*Tarık Gören, Mustafa Kahveci*
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
*200202022@kocaeli.edu.tr, 200202036@kocaeli.edu.tr*

## ÖZET

Bu projede yoğun indeks (dense index) yapısından faydalanarak öğrencilerin numara, ders kodu ve puan bilgilerini veri dosyasında (.dat) saklayan ve sonrasında bu verilerin öğrenci numaralarına göre indeks dosyasına (.txt) indeksleyen ve bu indeksleme yapısının kazandırmış olduğu bazı verimlilikler sayesinde projede istenen fonksiyonların disk erişimini minimuma indiren bir öğrenci kayıt programı gerçekleştirmemiz istenmiştir.

## GİRİŞ

Bu çalışmada bizden öğrencilerin numara, ders kodu ve puan bilgilerini veri dosyasında saklayan ve sonrasında bu verilerin öğrenci numaralarına göre indeks dosyasına küçükten büyüğe doğru sıralı bir şekilde indeksleyip verilere erişirken öğrenci numaraları anahtar olarak belirlenip kayıt arama, silme,güncelleme gibi fonksiyonların dense index yapısından faydalanarak bir öğrenci kayıt sistemi tasarlamak amaçlanmıştır.
Bu proje kapsamında yoğun indeks (dense index) yapısı gerçekleştirilecektir. Bu indeks yapısında ikili dosya (binary file) olarak gerçekleştirilen veri dosyasındaki her kayıt için indeks dosyasında bir girdi saklanmaktadır. İndeks dosyasında kayıtlar <anahtar adres> şeklinde saklanmaktadır. (**Şekil 1.1**)

**Şekil 1.1**

| Anahtar | Adres |
|---|---|
| 1 | 0 |
| 1 | 4x |
| 2 | x |
| 2 | 3x |
| 2 | 6x |
| 3 | 2x |
| . | |
| . | |
| . | |

Adres değeri anahtarla ilgili bilginin veri dosyasında hangi offsette saklandığını göstermektedir. (**Şekil 1.2**)

**Şekil 1.2**

| offset | Anahtar |
|---|---|
| 0 | 1 |
| x | 2 |
| 2x | 3 |
| 3x | 2 |
| 4x | 1 |
| 5x | . |
| 6x | 2 |
| 7x | . |
| 8x | . |

Veri dosyasında aynı anahtarla ilgili birden çok kayıt olabileceği için indeks dosyasında aynı anahtara ait birden fazla kayıt bulunacağı ve bu anahtarların sıralanmasının da bu yapıya uygun şekilde Şekil 1.1'de olduğu gibi tasarlanması istenmiştir.

Bizden programımızda bulunması istenen olan bazı fonksiyonlar ve işlevleri:

- indexDosyasiOlustur: Bir veri dosyasının henüz indekslenmediğini kabul edip veri dosyasındaki kayıtlar için bir indeks dosyası oluşturulacaktır.
- kayitEkle: Veri dosyasına yeni bir kayıt eklenecektir. Yeni kayıt her zaman veri dosyasının sonuna eklenmelidir. Kayıt eklendikten sonra indeks dosyası güncellenmelidir.
- kayitBul: Verilen bir anahtar için veri dosyasındaki ilk kayıt gösterilecektir. İlgili anahtar için birden çok kayıt bulunabileceği için aranılan kayıt bulunana kadar anahtara ait kayıtlar listelenecektir. İndeks dosyasında anahtara ait ilk kaydın bulunmasında ikili arama (binary search) algoritması kullanılacaktır.

• kayitSil: Verilen bir anahtar için ilgili doğru kayıt bulunacak (kayitBul işlemindeki gibi) ve bu kayıt silinecektir. Silme işlemi indeks dosyasına da yansıtılacaktır.
• kayitGuncelle: Bir anahtar için aranılan kayıt bulunup sadece puan alanı güncellenecektir.
• veriDosyasiniGoster: Veri dosyasındaki tüm kayıtları listeler.
• indeksDosyasiniGoster: İndeks dosyasındaki tüm kayıtları gösterir.
• indeksDosyasiniSil: İndeks dosyasını diskten siler.

İhtiyaç duyup kendi eklediğim fonksiyonlar ve işlevleri:

• diziBoyutBul: Veri dosyasında anlık ne kadar kayıt olduğunu bulur ve static bir int değişkene atar.
• guncelleVeSil: kayitSil ve kayitGuncelle fonksiyonlarını gönderilen puan değerine göre gerçekleştirir.
• binarySearch: İkili arama algoritmasını gerçekleştirir.
• indexDosyasiSirala: Index dosyasını sıralamaya yarar.

Bu çalışmada bizden standart C dilini kullanmamız istenmiştir.

## YÖNTEM

Öğrenci kayıt programımız için öncelikle bir menü oluşturuldu (switch case ile). Menü aracılığı ile fonksiyonlara erişim sağlandı. Kayıtlar için kayit tipinde 'kullanici' struct'ında alınıp kayitEkle fonksiyonunda veri dosyasında alt alta eklenecek şekilde veri dosyasına ekleme yaptırılmıştır. Her kayıt ekleme işleminden sonra eğer index dosyası daha önceden oluşturulmuş ise index dosyası güncellenmiştir. Eklenen kayıtları indekslemek için Ana Menüye "Index Dosyası Olustur" seçeneği eklenmiştir. Eğer index dosyası oluşturulmamışsa menüdeki kayıt güncelleme, kayıt silme, kayıt bulma, index dosyası yazdırma ve index dosyası silme bölümleri çalışmayacaktır. (Ana menüye yönlendirecektir.). Çünkü verilere erişim index dosyası aracılığı ile gerçekleşmektedir. Index dosyası oluşturma işlemi gerçekleştirilirken öncelikle kayıt sayısını hesaplaması için diziBoyutBul fonksiyonu çağrılmıştır. (Bu fonksiyon veri dosyasındaki kayıtları baştan sona okuyup kaç tane olduğunu bulmaya yarar ve bulduğu sayıyı **static** diziBoyut değişkenine eşitler.) Daha sonra indexFile.txt yazma modunda oluşturulur. Daha önceden **static** olarak oluşturduğumuz **int** *anahtarList ve **int** *adresList ve **struct** kayit *ogrenci işaretçilerine bellekte yer tahsisi yapılmıştır (diziBoyut kadar). Daha sonra veri dosyası baştan sona okunarak ogrenci struct'ına atılmıştır. Aynı zamanda anahtarList dizisine öğrenci numaraları, adresList dizisine de öğrenci numaralarının bulunduğu adres saklanmıştır. Sonra indexDosyasiSirala fonksiyonu çağırılarak bu dizilerin sıralanıp indexFile.txt'e yazdırılması sağlanmıştır. Sıralama algoritması olarak insertion sort kullanılmıştır. Sıralı dizi siraliOgrNoArray'de saklanmıştır. İç içe döngü kullanarak sıralıogrNoArray ve anahtarList(sırasız) karşılaştırması yapılmıştır. Eşitse index dosyasına anahtarların karşısına doğru adres yazdırılmıştır ve sırasız olan anahtarList'teki ilgili indis -1'e eşitlenmiştir(Bir sonraki kontrolde aynı anahtara sahip kaydın adresini görmemesi için). Döngü tamamlandıktan sonra index dosyası sıralı ve doğru adresler karşısında bulunacak bir şekilde sıralanmıştır. Sıralama fonksiyonunun sonunda index dosyasındaki anahtar ve adresler tekrardan okunarak anahtarList ve adresList'e aktarılmıştır. kayitBul(**FILE** *dataPtr, **int** silAnahtar) fonksiyonunda ise öncelikle indexDosyasiOlustur fonksiyonunu çalıştırılmıştır.(index dosyasını ve static olarak tanımladığımız değişkenleri güncellemek için). Eğer silAnahtar parametresi -1 girilmiş ise kullanıcıdan aranacak öğrenci numarası istenir. Eğer -1'den farklı ise silAnahtar arananOgrNo değişkenine eşitlenir. Bunun sebebi kayıt bul fonksiyonunu kayitSil ve kayitGuncelle fonksiyonlarından çağırmışmıyız bunun kontrolünü sağlamak. Daha sonra arananIndis değişkenini binarySearch fonksiyonuna gönderdiğimiz anahtarın anahtarList'te aranması ile geri dönen indise eşitliyoruz. Aynı anahtara ait birden fazla kayıt olma durumunda bazen binarySearch fonksiyonu bize tam olarak istediğimiz indisi veremeyebiliyor. Örneğin 5 tane 1001010 numaralı öğrenci kaydı

olsun.Binary search yaptığımız dizinin boyutuna göre bazen 3. veya 4. indisi döndürebiliyor ve bunun sonucunda (3. indis dönmüş varsayalım) ekrana veri dosyasındaki baştan 3. 4. ve 5. kayıtları ekrana yazdırıp 1 ve 2. es geçebiliyor. Bunun kontrolü için binarySearch ile dönen ilk indisten önceki indislerin aradığımız anahtara eşit mi diye kontrolünü bir while döngüsünde aratıyoruz. Eğer eşitse arananIndis birer birer azaltılıyor ve ilk kaydın indisi bulunmuş oluyor. Daha sonra ilgili anahtarın ders kaydı kadar döngüye girip fonksiyonun başında oluşturulan ptr değişkenine (struct kayit) adresListteki ilgili adres ataması yapılıyor ve struct pointerının işaret ettiği kayıt ekrana yazdırılıyor. KayitSil fonksiyonunda kullanıcıdan ilk olarak silinmesi istenilen öğrenci numarası alınıyor. Daha sonra kontrol adlı değişken kayitBul fonksiyonunu kullanıcıdan alınan anahtar ile parametre olarak çağırılarak eşitleniyor. Geriye dönen sayı -1'e eşitse kayıt bulunamadığı için yeniden bir girdi isteniyor. Eğer eşit değilse de girilen öğrenci numarasına ait ilgili ders kodunun girilmesi isteniyor.(**int** silDersKodu) Güncelleme ve silme işlemlerinin mantığı birbirine benzer olduğu için guncelleVeSil adlı fonksiyonda toparlanmıştır. Kullanıcıdan almış olduğumuz silNo ve silDersKodu parametrelerini guncelleVeSil fonksiyonuna gönderiyoruz. Güncelleme işleminden farklı olarak girilenPuan parametresini -1 olarak giriyoruz. GuncelleVeSil fonksiyonunun silme bölümünde veri dosyası okuma modunda en baştan itibaren okunuyor. Eğer göndermiş olduğumuz parametredeki kayıt okunduysa o kaydı yazdırmadan devam ediyor okunmamışsa diğer kayıtları alt alta olacak şekilde temp.txt adlı dosyaya yazıyor. Daha sonra dataFile.dat dosyamız silinip temp.dat dosyamız (güncel veri dosyası)'ın ismi dataFile.dat olarak değiştiriliyor ve kayıt silme işlemi tamamlanmış oluyor. Aynı mantıkta kayitGuncelle fonksiyonunda girdiler alınıyor guncelleVeSil fonksiyonuna gönderiliyor bu sefer güncelleme yapacağımız için kayıt silmeden farklı olarak gönderdiğimiz parametredeki kaydın okunduğu sırada temp.dat dosyasına yazdırırken sadece puan bölümünü girilenPuan'a eşitlenerek güncelleniyor. Devamındaki işlemler de silme ile aynıdır. Her kayıt silme ve güncelleme işlemlerinden sonra da index dosyası güncellenmesi için indexDosyasiOlustur fonksiyonu çağırılıyor.

## SONUÇLAR

Bizden istenen tüm isterler gerçekleştirilmiştir. Kullanıcıdan alınan öğrenci bilgi verileri doğru bir şekilde veri dosyasına yazdırıldı.Yazdırılan verilerin indeks dosyasında yoğun indeks yapısı kullanarak indekslenmesi sağlandı. Daha sonra indeks dosyası aracılığıyla istenen verilere erişim sağlandı. Proje sonucunda dense index yapısının kullanımı ve mantığı, dinamik bellek tahsisi, ikili arama algoritması ile daha çabuk bir şekilde sıralı bir dizide arama yapılması ve dosya işlemlerinde kullanılan çeşitli fonksiyonların kullanımı öğrenildi. Proje CodeBlocks'ta kodlanmıştır. Gerekli testler yapılarak programın runtime hataları vermesi önlendi.

## KAYNAKLAR


• Onur Gök ve Suhap Sahin Veri Yapıları ve Algoritmalar Dersi Slaytları

•https://web.cs.hacettepe.edu.tr/~maydos/Docs/c/dosyalar.pdf

•https://www.geeksforgeeks.org/dynamic-memory-allocation-in-c-using-malloc-calloc-free-and-realloc/

•https://bilgisayarkavramlari.com/2009/12/21/ikili-arama-algoritmasi-binary-search-algorithm/

• Onur Gök ve Suhap Sahin Programlama Dersi Slaytları

•https://www2.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter11/node5.html

•http://mlwiki.org/index.php/Dense_Index

•https://www.youtube.com/watch?v=2QmXifFCcGk

•https://stackoverflow.com/questions/25381610/cast-int-to-pointer-why-cast-to-long-first-as-in-p-void-42/25383329

# ÇIKTILAR

## Kayıt Ekleme

```
Ana Menu
'1' - Kayit Ekle
'2' - Kayit Guncelleme
'3' - Kayit Sil
'4' - Kayit Bul
'5' - Index Dosyasi olustur
'6' - Veri Dosyasini Yazdir
'7' - Index Dosyasini Yazdir
'8' - Index Dosyasini Sil
 (Programi sonlandirmak icin '-1')
 1
Kac adet? : 1

Ogrenci No - Ders Kodu - Puan
200202022 1 95
```

## Kayıt Güncelleme

```
Guncellemek istediginiz ogrencinin no giriniz : 200202022

        Ogrenci No              Ders Kodu       Puan
        200202022               2               100
        200202022               5               100
        200202022               1               95
+ Guncellemek istediginiz kaydin ders kodunu girin:1

Yeni puan degeri giriniz: 75

+ Kayit guncelleme islemi basariyla tamamlandi!
```

## Kayıt Silme

```
Silmek istediginiz ogrenci no giriniz: 200202022

        Ogrenci No              Ders Kodu       Puan
        200202022               2               100
        200202022               5               100
        200202022               1               75
+ Silmek istediginiz kaydin ders kodunu girin:1

+ Kayit silme islemi basariyla tamamlandi!
```

## Kayıt Bulma

```
Aradiginiz ogrencinin numarasini giriniz:200202022

        Ogrenci No              Ders Kodu       Puan
        200202022               2               100
        200202022               5               100
```

## Veri Dosyası Gösterme

```
 Ogrenci No      Ders Kodu       Puan
 200202022               2               100
 200202028               1               69
 200202023               1               92
 200202030               4               67
 200202031               5               59
 200202016               2               95
 200202008               5               84
 200202006               4               35
 200202041               1               72
 200202001               3               70
 200202004               1               65
 200202037               4               75
 200202003               2               20
 200202005               4               80
 200202038               2               75
 200202008               1               68
 200202026               3               62
 200202014               1               70
 200202002               3               55
 200202040               4               88
 200202007               3               50
 200202001               1               65
 200202004               3               75
 200202018               3               76
 200202007               2               70
 200202015               5               75
 200202029               1               83
 200202001               2               55
 200202021               2               64
 200202013               2               85
 200202007               5               80
 200202006               1               75
 200202025               5               94
 200202027               2               87
 200202009               2               65
 200202033               2               79
 200202017               1               65
 200202005               2               66
 200202019               4               81
 200202035               3               89
 200202024               3               86
 200202032               1               74
 200202010               1               80
 200202011               2               85
```

## Index Dosyası Gösterme

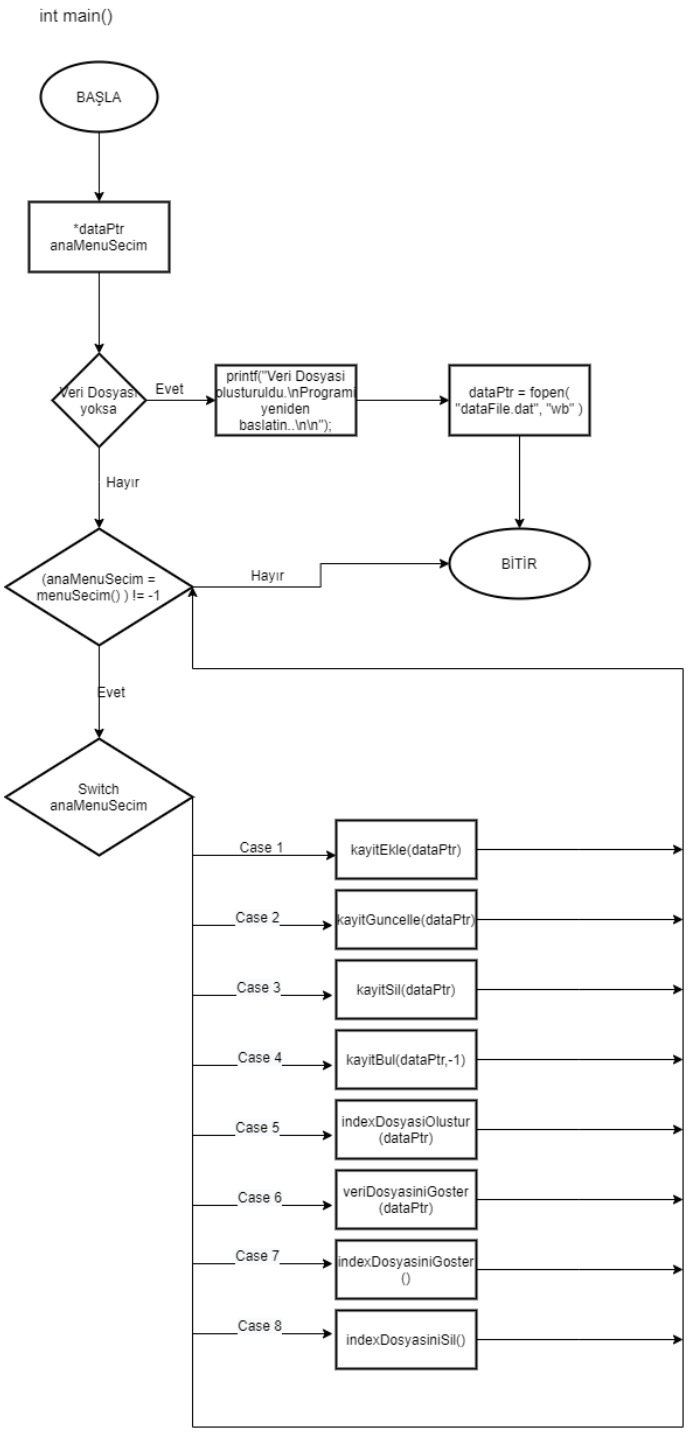
```
Anahtar         Offset
200202001               8102812
200202001               8102956
200202001               8103028
200202002               8102920
200202003               8102848
200202004               8102824
200202004               8102968
200202005               8102860
200202005               8103148
200202006               8102788
200202006               8103076
200202006               8103268
200202007               8102944
200202007               8102992
200202007               8103064
200202008               8102776
200202008               8102884
200202009               8103112
200202010               8103208
200202011               8103220
200202012               8103232
200202013               8103052
200202014               8102908
200202015               8103004
200202016               8102764
200202017               8103136
200202018               8102980
200202019               8103160
200202020               8103256
200202021               8103040
200202022               8102704
200202022               8103304
200202023               8102728
200202024               8103184
200202025               8103088
200202026               8102896
200202027               8103100
200202028               8102716
200202029               8103016
200202030               8102740
200202031               8102752
200202032               8103196
200202033               8103124
200202034               8103280
200202035               8103172
```

## Index Dosyası Silme

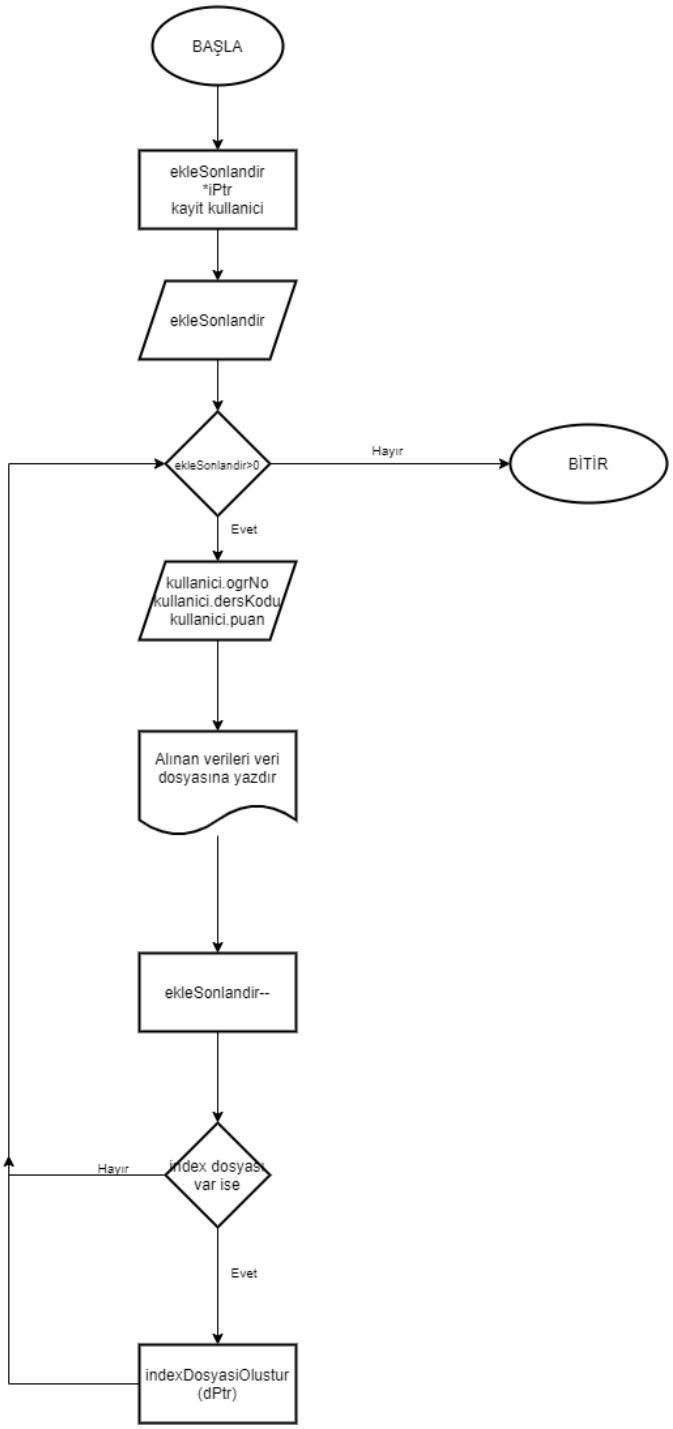
```
+ Index dosyasi diskten basariyla silindi.

Ana Menu
'1' - Kayit Ekle
'2' - Kayit Guncelleme
'3' - Kayit Sil
'4' - Kayit Bul
'5' - Index Dosyasi olustur
'6' - Veri Dosyasini Yazdir
'7' - Index Dosyasini Yazdir
'8' - Index Dosyasini Sil
 (Programi sonlandirmak icin '-1')
 2

- Veri dosyasi indekslenmemis.Indeks dosyasi olusturun!
```

## AKIŞ ŞEMALARI

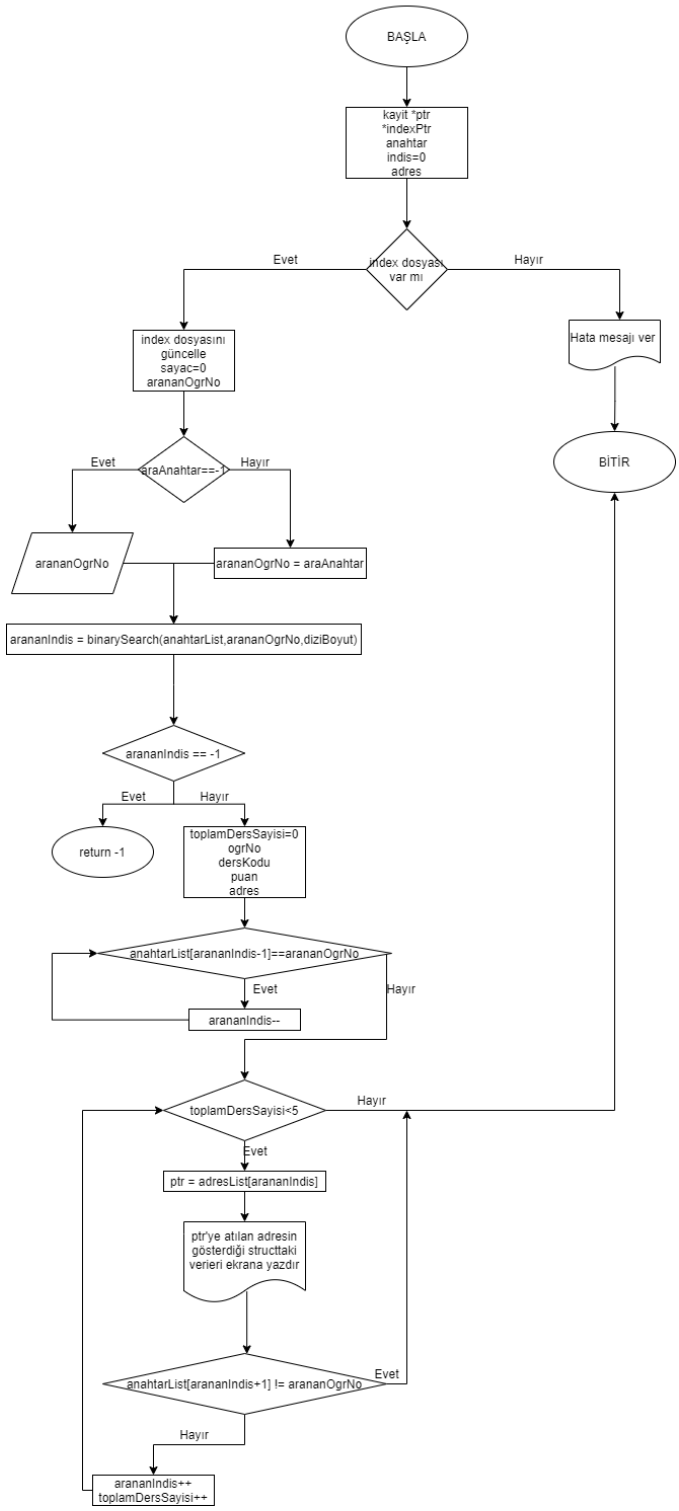
Akış şemaları draw.io üzerinden oluşturulmuştur.

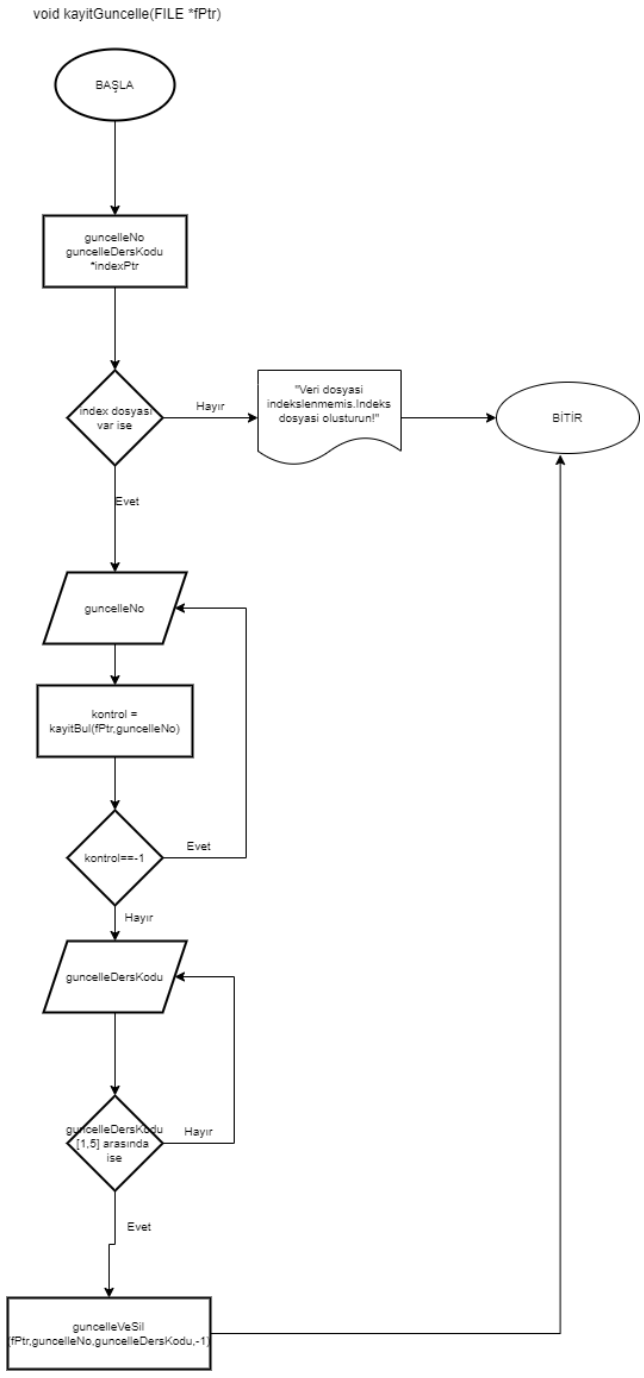


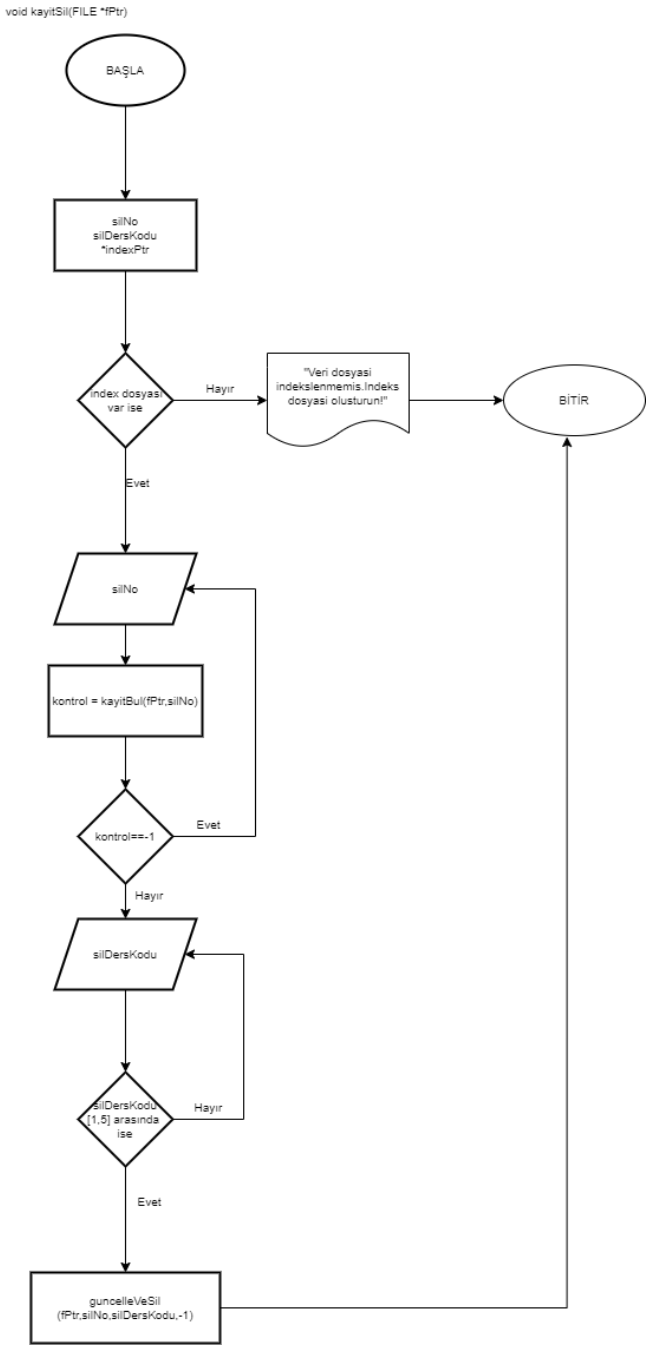

void kayitEkle(FILE *dPtr)



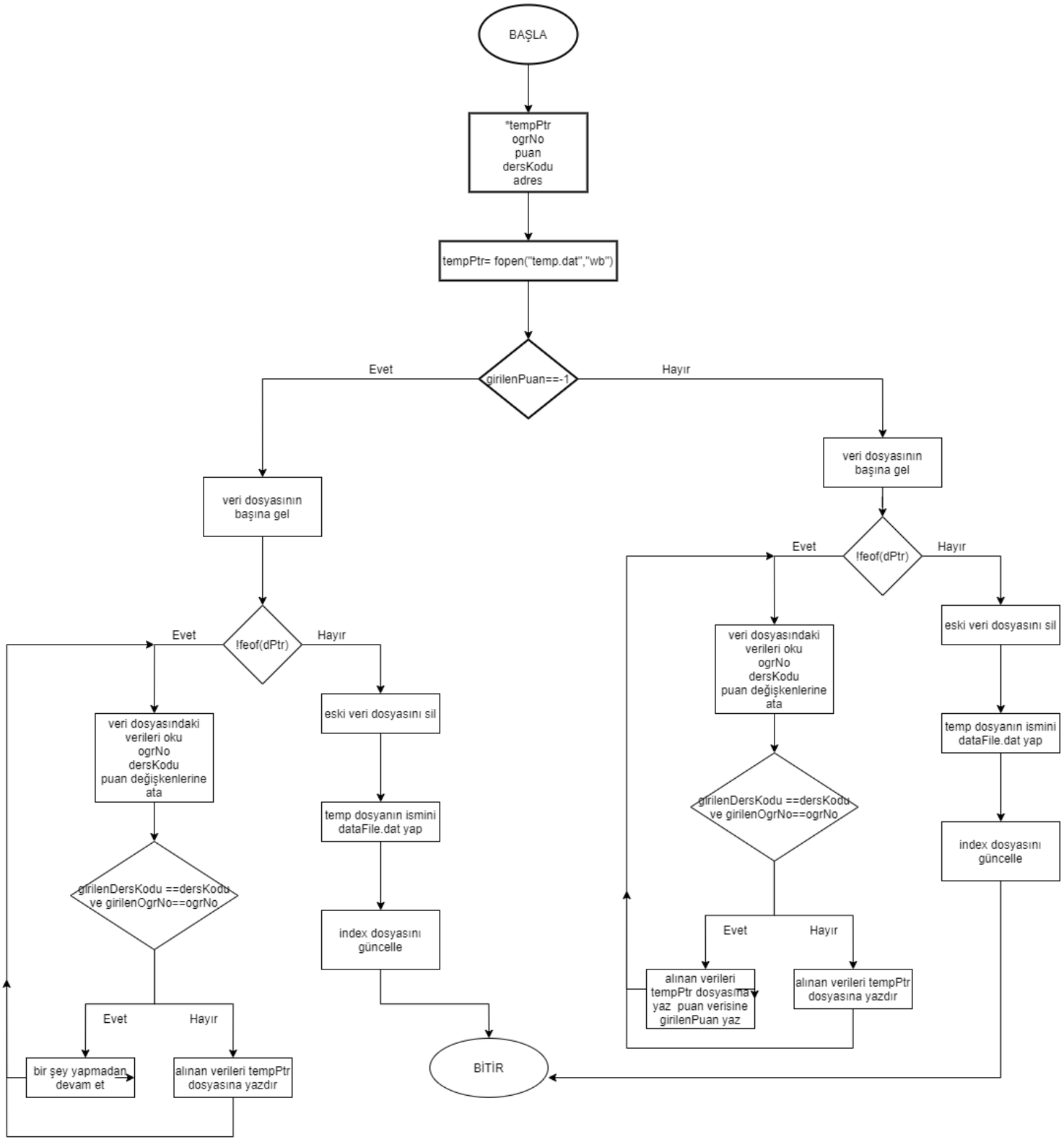

int kayitBul(FILE *dataPtr,int araAnahtar)

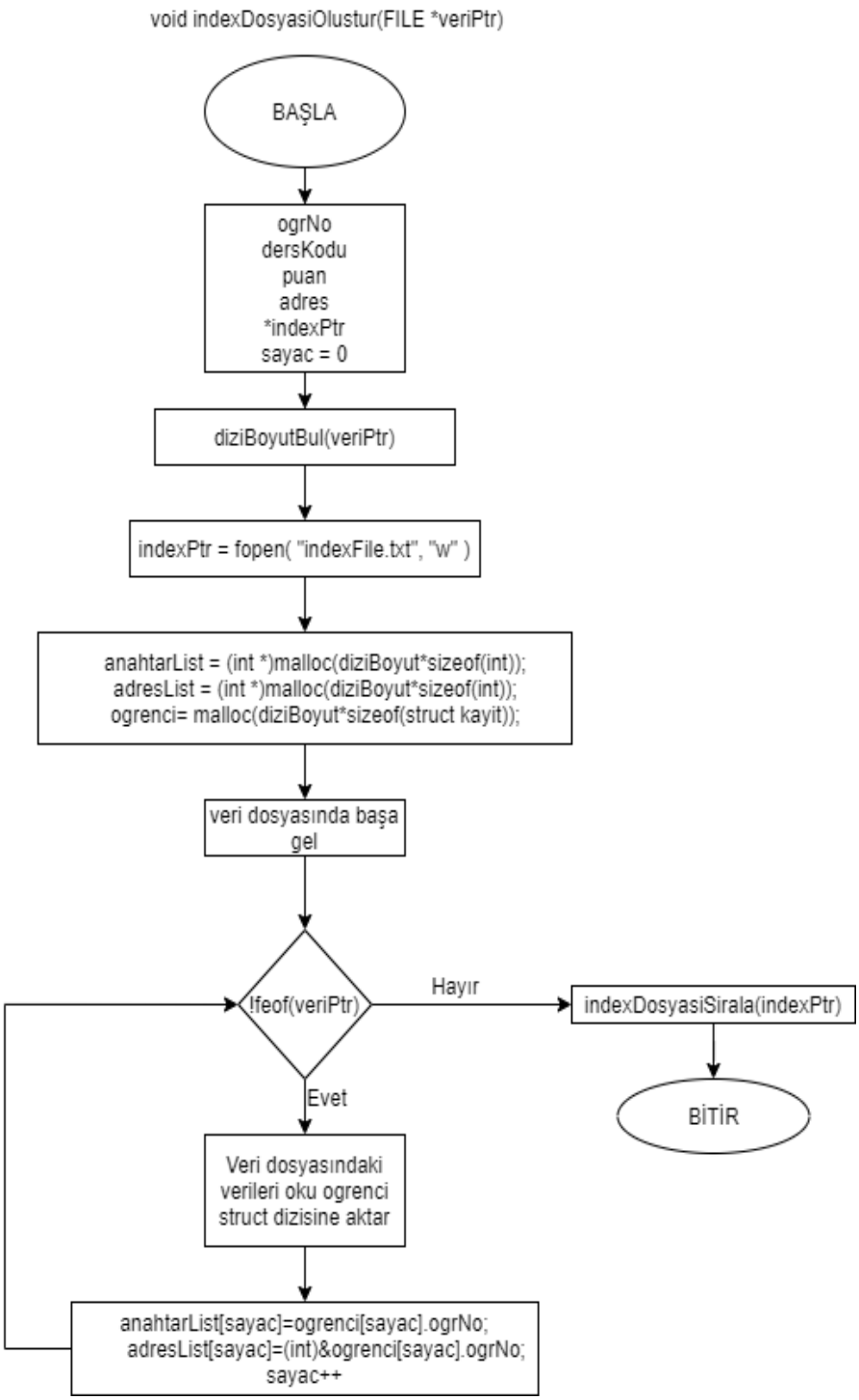


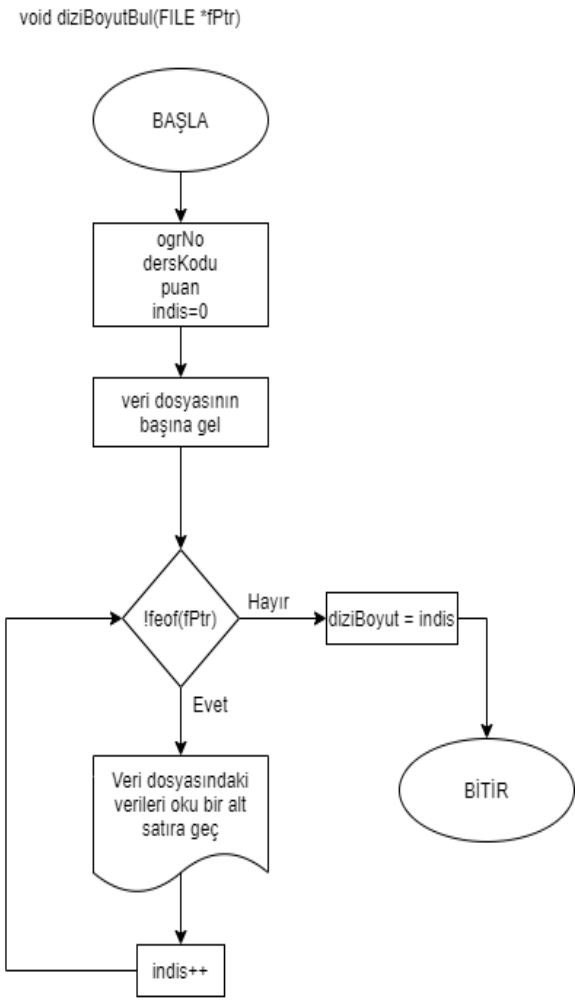

void kayitGuncelle(FILE *fPtr)
BAŞLA
guncelleNo
guncelleDersKodu
*indexPtr
index dosyasi
var ise
Hayır
"Veri dosyasi
indekslenmemis.Indeks
dosyasi olusturun!"
BİTİR
Evet
guncelleNo
kontrol =
kayitBul(fPtr,guncelleNo)
kontrol==-1
Evet
Hayır
guncelleDersKodu
guncelleDersKodu
[1,5] arasında
ise
Hayır
Evet
guncelleVeSil
(fPtr,guncelleNo,guncelleDersKodu,-1)

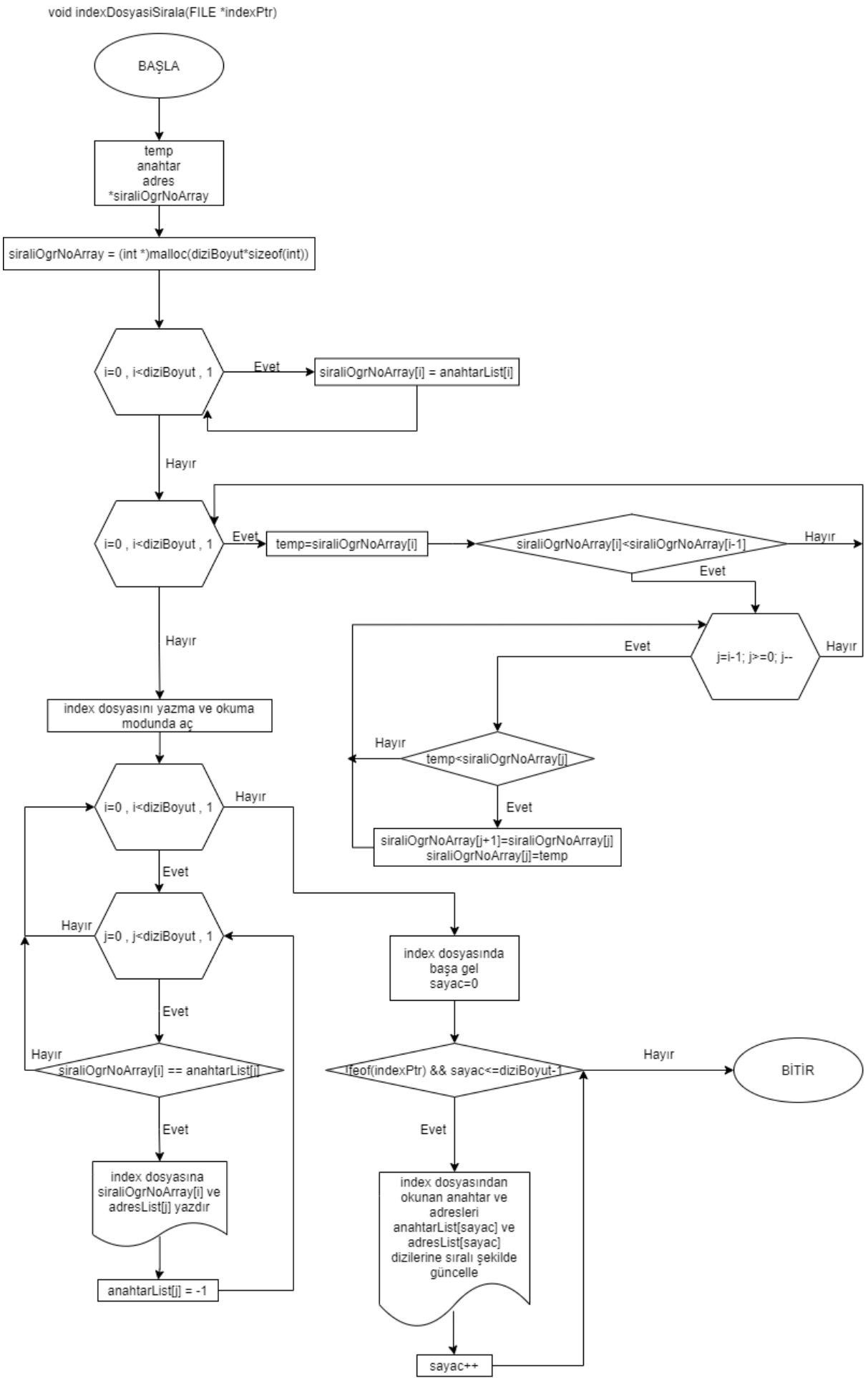

void kayitSil(FILE *fPtr)
BAŞLA
silNo
silDersKodu
*indexPtr
index dosyasi var ise
Hayır
"Veri dosyasi indekslenmemis.Indeks dosyasi olusturun!"
BİTİR
Evet
silNo
kontrol = kayitBul(fPtr,silNo)
kontrol==-1
Evet
Hayır
silDersKodu
silDersKodu [1,5] arasında ise
Hayır
Evet
guncelleVeSil (fPtr,silNo,silDersKodu,-1)

void guncelleVeSil(FILE *dPtr,int girilenOgrNo,int girilenDersKodu,int girilenPuan)

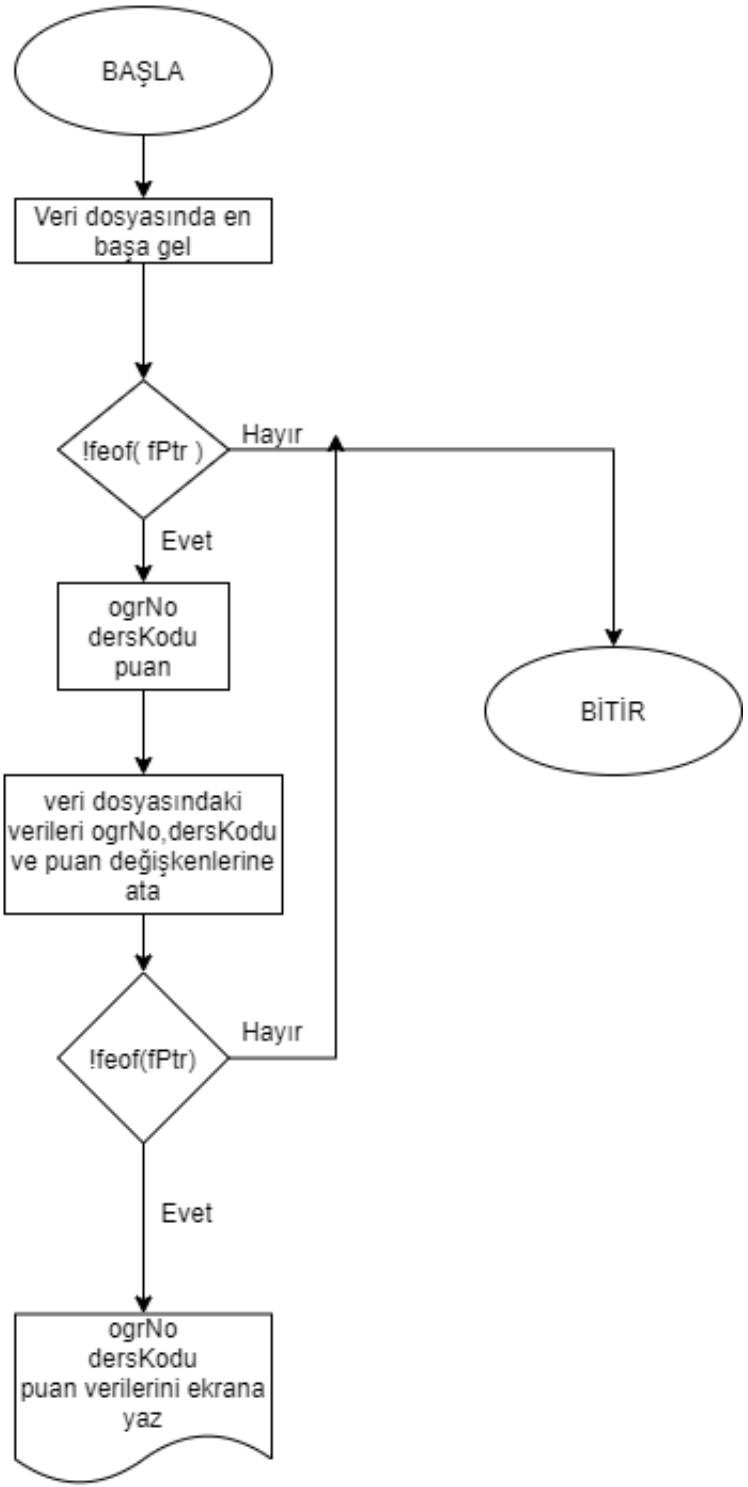



void indexDosyasiOlustur(FILE *veriPtr)
BAŞLA
ogrNo
dersKodu
puan
adres
*indexPtr
sayac = 0
diziBoyutBul(veriPtr)
indexPtr = fopen( "indexFile.txt", "w" )
anahtarList = (int *)malloc(diziBoyut*sizeof(int));
adresList = (int *)malloc(diziBoyut*sizeof(int));
ogrenci= malloc(diziBoyut*sizeof(struct kayit));
veri dosyasında başa gel
!feof(veriPtr)
Hayır
indexDosyasiSirala(indexPtr)
BİTİR
Evet
Veri dosyasındaki verileri oku ogrenci struct dizisine aktar
anahtarList[sayac]=ogrenci[sayac].ogrNo;
adresList[sayac]=(int)&ogrenci[sayac].ogrNo;
sayac++

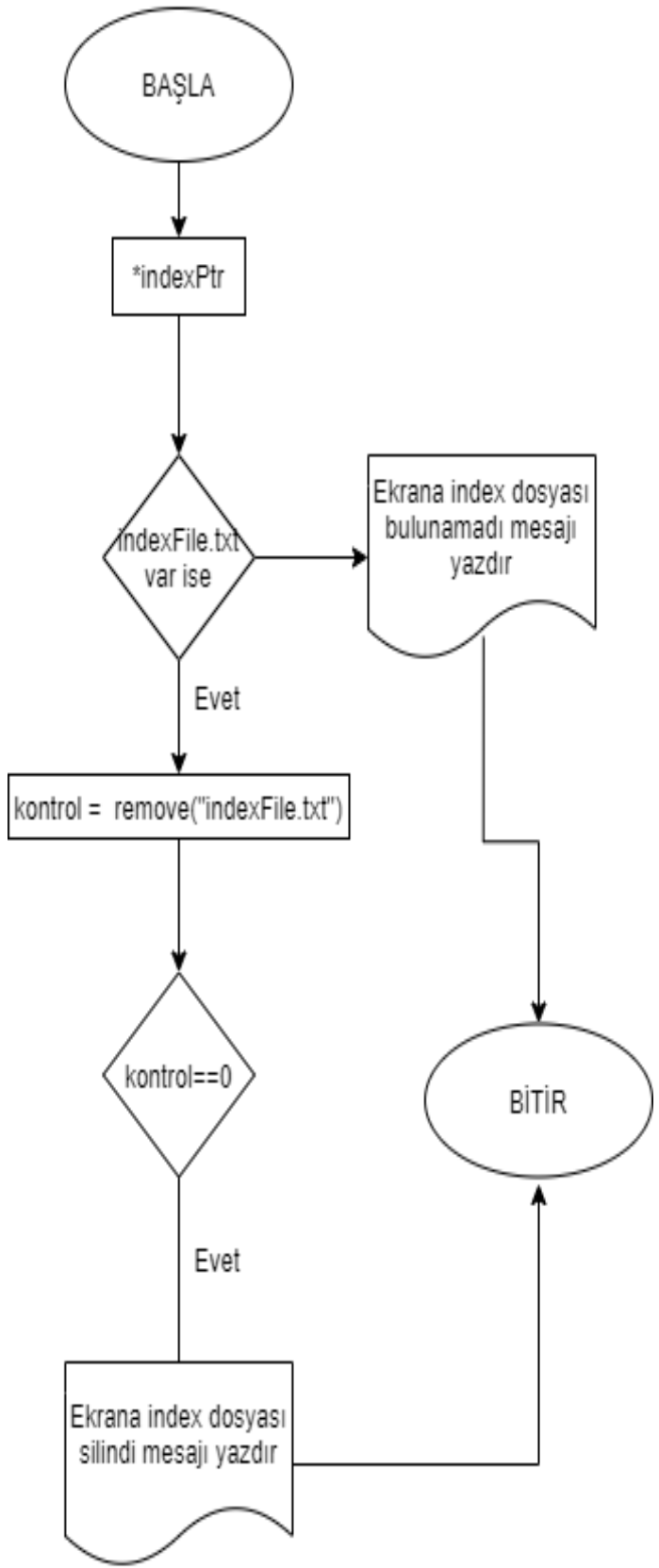

void diziBoyutBul(FILE *fPtr)
BAŞLA
ogrNo
dersKodu
puan
indis=0
veri dosyasının başına gel
!feof(fPtr)
Hayır
diziBoyut = indis
BİTİR
Evet
Veri dosyasındaki verileri oku bir alt satıra geç
indis++


void indexDosyasiSirala(FILE *indexPtr)
BAŞLA
temp
anahtar
adres
*siraliOgrNoArray
siraliOgrNoArray = (int *)malloc(diziBoyut*sizeof(int))
i=0 , i<diziBoyut , 1
Evet
siraliOgrNoArray[i] = anahtarList[i]
Hayır
i=0 , i<diziBoyut , 1
Evet
temp=siraliOgrNoArray[i]
siraliOgrNoArray[i]<siraliOgrNoArray[i-1]
Hayır
Evet
Hayır
j=i-1; j>=0; j--
Evet
Hayır
temp<siraliOgrNoArray[j]
Hayır
Evet
siraliOgrNoArray[j+1]=siraliOgrNoArray[j]
siraliOgrNoArray[j]=temp
index dosyasını yazma ve okuma modunda aç
i=0 , i<diziBoyut , 1
Hayır
Evet
j=0 , j<diziBoyut , 1
Hayır
Evet
siraliOgrNoArray[i] == anahtarList[j]
Hayır
Evet
index dosyasına siraliOgrNoArray[i] ve adresList[j] yazdır
anahtarList[j] = -1
index dosyasında başa gel
sayac=0
!feof(indexPtr) && sayac<=diziBoyut-1
Hayır
BİTİR
Evet
index dosyasından okunan anahtar ve adresleri anahtarList[sayac] ve adresList[sayac] dizilerine sıralı şekilde güncelle
sayac++

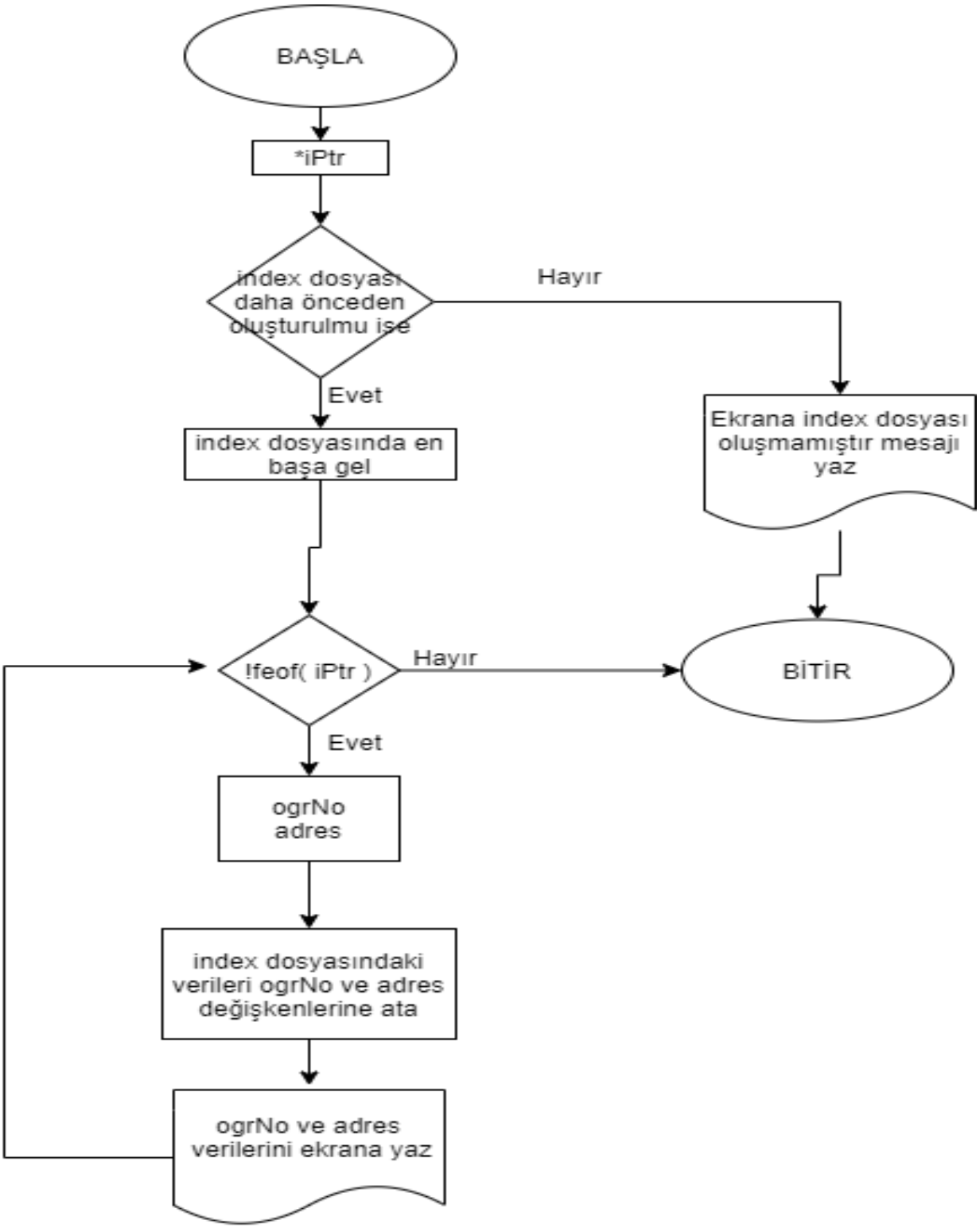
void veriDosyasiniGoster(FILE *fPtr)




void indexDosyasiniSil()



indexDosyasiniGoster()



int binarySearch(int *arr, int anahtar,int diziBoyut)