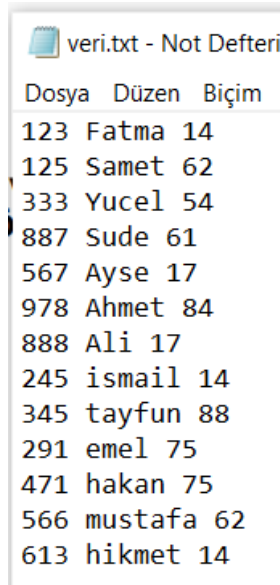


### Veri Yapıları Ödev 3 (Hash Veri Yapısı Uygulaması)

```
typedef struct {  
    int ogr_no;  
    char ad[20];  
    int notu  
} Ogrenci;
```



Dosya	Düzen	Biçim
123	Fatma	14
125	Samet	62
333	Yucel	54
887	Sude	61
567	Ayşe	17
978	Ahmet	84
888	Ali	17
245	ismail	14
345	tayfun	88
291	emel	75
471	hakan	75
566	mustafa	62
613	hikmet	14

Yukarıdaki yapıda olan ve bilgileri “veri.txt” dosyasında tutulan kayıtlar öğrenci notuna göre indekslenecektir (satırlardaki 3. değer). Öğrenci bilgileri için “Ogrenci” veri türünde 10 boyutunda bir **veri\_liste** dizisi oluşturulacaktır.

Ogrenci veri\_liste[10];

Veri eklendikçe oluşturulan “Ogrenci” türündeki yapının adresi veri\_liste dizisindeki, aşağıdaki gibi hesaplanan (hash) indeksindeki elemana atanacaktır.

Hash tablosu için boyutu 10 olan **int** türünde ayrı bir dizi olan **hash\_tablo** dizisi oluşturulacaktır. Hash\_tablo dizisinde öğrencinin notunun haricinde çakışma olması durumunda aranacak hash indeksini gösteren bir **next** değeri tutulacaktır. Hash\_tablo’da **başlangıçta bütün değerler (notu ve next) “-1” olacaktır.**

**int** hash\_tablo[10][2];

Tabloda tutulacak değerler

hash\_tablo[x][0]----->notu  
hash\_tablo[x][1]-----> next

Buna göre:

**Ekleme İşlemi:** Ekleme işlemi “veri.txt” dosyasındaki kayıtlar üzerinden ve ekrandan kullanıcı tarafından yapılabilecektir. Program ilk açıldığında “veri.txt” dosyasındaki kayıtlar için aşağıda açıklandığı gibi **hash\_tablo** oluşturulacak ve veriler hash tablosundaki indeksine göre **veri\_liste** dizisine eklenecektir.

Yazacağımız hash fonksiyonu öğrenci notunun 10’a bölümüne kalan değer (**notu % 10**) olarak belirlenecektir ve hash\_tablo’daki o indekse öğrencinin notu kaydedilecektir (**çakışma yoksa next değeri -1 olarak kalacaktır**).

Örneğin öğrencinin notu 14 ise hash indeksi 4 olacaktır.

Hash tablosuna her not eklendiğinde, öğrenci bilgileri de veri\_liste’nin aynı indeksine eklenecektir.

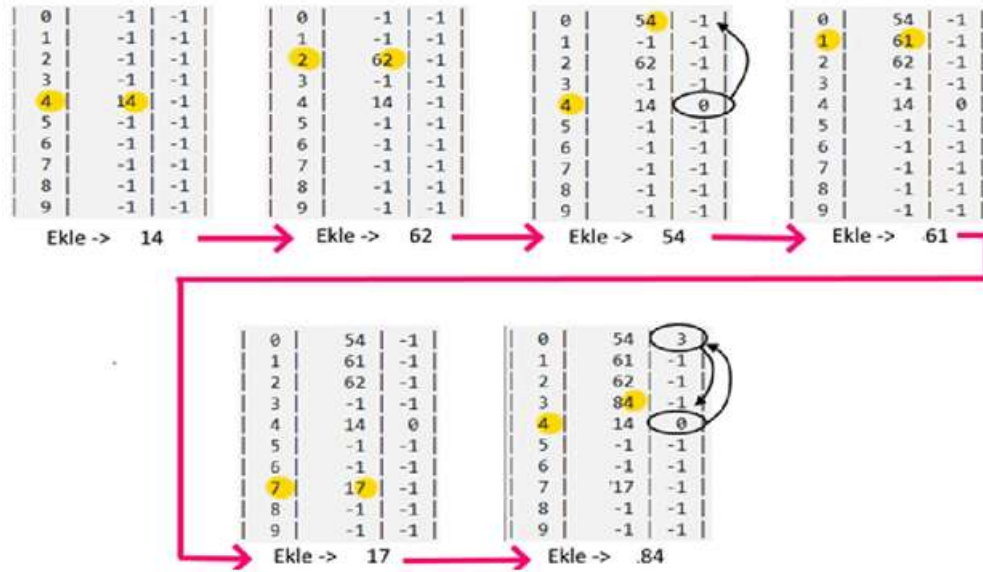
hash_tablo	
-1	-1
-1	-1
-1	-1
-1	-1
14	-1
-1	-1
-1	-1
-1	-1
-1	-1
-1	-1

veri_liste
NULL
NULL
NULL
NULL
{ 123,Fatma,14 }
NULL
NULL
NULL
NULL
NULL

Eğer bir çakışma olursa hash\_tablo'da en baştan ilk bulunduğu boş yere kaydedilecektir ve aldığı indeks değerini çakışmanın olduğu indeksteki next değerini gösteren alana ekleyecektir. Eğer burada daha önceden çakışma olup başka bir next değeri yazıldıysa (-1 değilse) daha önce girilmiş olan next numarasını hash\_tablo'dan takip edecek ve ilk ulaştığı next değeri -1 olan alana, aldığı indeksin değerini yazacaktır. **Hash Tablosu dolduğunda “tablo dolu” mesajı verilerek daha fazla ekleme yapılamayacaktır.**

**Örnek 1:** Aşağıdaki şekilde 54 eklendiğinde 4. indeks dolu olduğu (notu -1 olmadığı) için ilk boş bulunduğu alan olan 0. indekse eklenmekte ve aldığı indeks değeri olan 0 değeri 4. indeksteki next alanına yazılmaktadır.

**Örnek 2:** Aşağıdaki şekilde 84 eklendiğinde 4. indeks dolu olduğundan ilk bulunduğu boş yer olan 3. indekse eklenmiştir. Aldığı indeks değeri olan 3 değerini yazmak için, 4. indeksin next alanı -1 olmadığından oraya daha önceden girilmiş olan 0. indekse gitmiştir ve oradaki next alanı -1 olduğu için kendi indeksi olan 3 değerini oraya yazmıştır.



**Arama İşlemi:** Arama işleminde, kullanıcı tarafından girilen öğrenci notundan hesaplanan (notu % 10) hash indeksindeki numara (hash\_tablo'daki) öğrenci notu ile aynı ise veri listesinin hesaplanan indeksteki bilgileri ekrana yazdırılacak ve **kaç adımda ulaştığı** (ilk baktığı yerde ise 1) ekrana yazdırılacaktır. Eğer öğrenci notu ile aynı değilse yandaki **next** değerlerini takip ederek öğrenci notunu bulmaya çalışacaktır (fonksiyon kaçınıcı adımda bulunduğunu döndürebilir). Eğer baktığı yer, **hem öğrenci notu ile aynı değil hem de next değeri -1** ise “Bulunamadı” olarak ekrana yazdırılacaktır (fonksiyon -1 değer döndürebilir).

```
Aranilan Deger: 84
84
Veri 3 adimda bulundu.
```

```
Aranilan Deger: 62
62
Veri 1 adimda bulundu.
```

```
Aranilan Deger: 63
Veri bulunamadi.
```

**Hash tablosu listeleme:** Oluřturulan hash\_tablo dizisi dzenli bir řekilde ekrana yazdırılacaktır.

54	3
61	-1
62	-1
84	-1
14	0
75	-1
-1	-1
17	-1
88	-1
-1	-1

Ödevde sorudaki işlemler için bir **ana menü** olacak ve her işlemten sonra ana menüye dönülecektir.

#### **NOT:**

Ödevler 26.12.2021 Pazar günü 23:55'e kadar <https://lms.ktun.edu.tr> adresi üzerinden uzaktan eğitim platformuna ogrenci\_numarasi\_Ad\_Soyad\_odev\_3.rar formatında yüklenecektir. Son teslim tarihinden sonra ödev sistemi otomatik kapanacak ve e-posta yolu ile gönderilecek ödevler kesinlikle dikkate alınmayacaktır. Kopya (doğrudan internet kaynağından alınan, arkadaştan alınarak deęişken ve fonksiyon ismi deęiřtirilen vb. her türlü kopya) ödevlere eksi puan verilecektir. Ödevler Final notuna etki edeceği için ödevleri yapmanız sizler için önem arz etmektedir.