

## Veri Yapıları Ödev 4 (Hash)

```
typedef struct {  
    int musteri_no;  
    char ad[20], soyad[20];  
} Musteri;
```



Yukarıdaki yapıda olan ve bilgileri "veri.txt" dosyasında tutulan kayıtlar müşteri numarasına göre indekslenecektir. Müşteri bilgileri için "Musteri\*" veri türünde 10 boyutunda bir **veri\_liste** dizisi oluşturulacaktır.

```
Musteri *veri_liste[10];
```

Veri eklendikçe oluşturulan "Musteri" 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 müşteri numarasının haricinde, çakışma olması durumunda aranacak hash indeksini gösteren bir **next** değeri tutulacaktır. Hash\_tablo'da **başlangıçta bütün değerler (müşteri\_no ve next) "-1" olacaktır.**

```
int hash_tablo[10][2];
```

### Tabloda tutulacak değerler

hash\_tablo[x][0] ---- > musteri\_no

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

Buna göre:

**1-Ekleme İşlemi:** Eklem 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.

Yazacağınız hash fonksiyonu müşteri numarasının 10'a bölümünden kalan değer (musteri\_no % 10) olarak belirlenecektir ve hash\_tablo'daki o indekse müşteri numarası kaydedilecektir (**çakışma yoksa next değeri -1 olarak kalacaktır**).

Örneğin müşteri numarası 2214 ise hash indeksi 4 olacaktır.

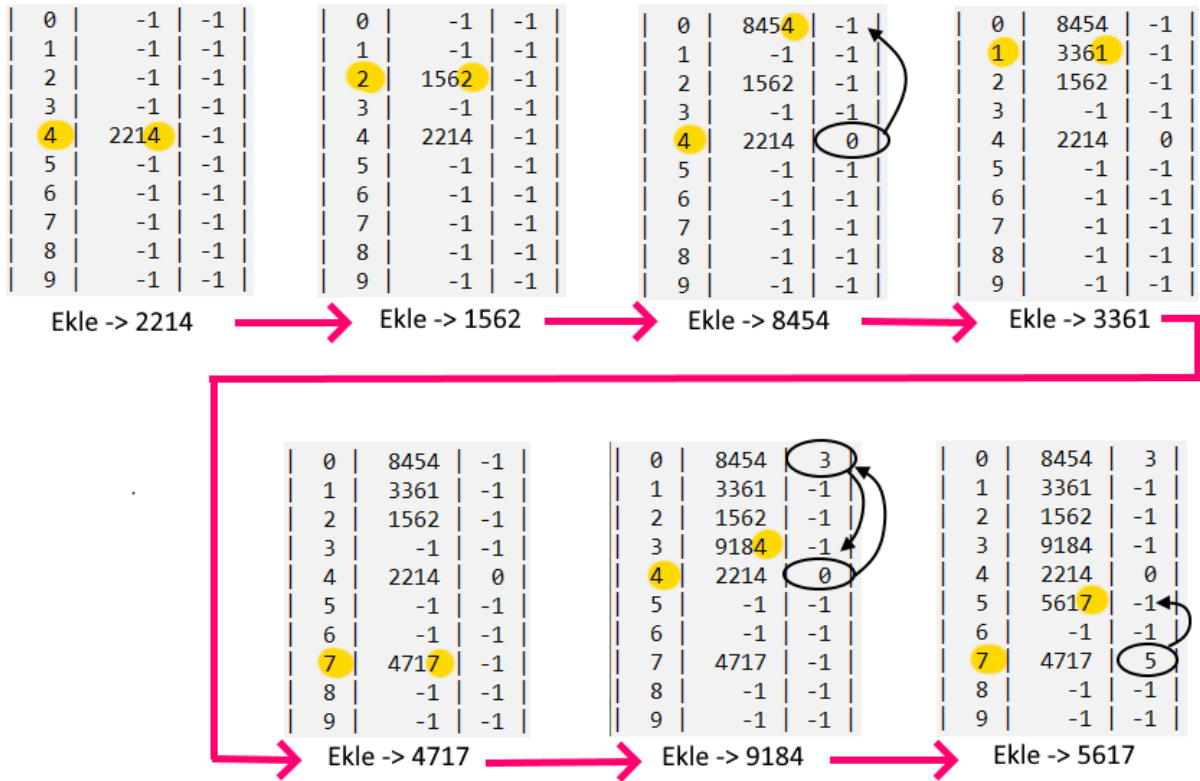
Hash tablosuna her müşteri numarası eklendiğinde, müşteri bilgileri de veri\_liste'nin aynı indeksine eklenecektir.

hash_tablo		veri_liste	
-1	-1	NULL	
-1	-1	NULL	
-1	-1	NULL	
-1	-1	NULL	
1234	-1	{1234, Ahmet, SAYAN}	
-1	-1	NULL	
-1	-1	NULL	
-1	-1	NULL	
-1	-1	NULL	
-1	-1	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 kayıt eklenemeyecektir.**

**Örnek 1:** Aşağıdaki şekilde 8454 eklendiğinde 4. indeks dolu olduğu (musteri\_no -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 9184 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.



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

Aranılan Deger: 9184

9184 Ahmet SURER  
Veri 3 adımda bulundu.

Aranılan Deger: 1562

1562 Samet CANSIZ  
Veri 1 adımda bulundu.

Aranılan Deger: 1563

Veri bulunamadi.

**3-Silme İşlemi:** Kullanıcının girdiği müşteri numarasına ait bilgiler hash\_tablo'dan ve veri liste'den silinecektir. Silinen verinin indeksi boş kalacağı için daha sonra başka bir veri eklenebilecektir. Silme işleminde, daha önceden diğer verilere ulaşılmasını sağlayan next değerleri değişmeyecektir.

**4-Hash tablosu listeleme:** Oluşturulan has\_tablo dizisi düzenli bir şekilde ekrana yazdırılacaktır.

```
8454 3
3361 -1
1562 -1
9184 -1
2214 0
5617 -1
-1 -1
4717 5
-1 -1
-1 -1
```

**5-Veri listesini listeleme:** Oluşturulan veri\_liste dizisine eklenen veriler dizideki sırasına göre ekranda düzenli bir şekilde listelenecektir.

```
8454 Yucel BAS
3361 Sude CIFTCI
1562 Samet CANSIZ
9184 Ahmet SURER
2214 Fatma AYDIN
5617 Ali SURUCU
4717 Ayse YILMAZ
```

**6-Ortalama Adım Sayısı:** Tüm kayıtlara (boş olmayan) **ortalama** kaç adımda ulaşıldığı ekrana yazdırılacaktır.

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

**Not:**

1-Ödevler <https://lms.ktun.edu.tr> adresi üzerinden uzaktan eğitim platformuna ödev 4 linki üzerinden **öğrenci\_numarasi\_Ad\_Soyad\_odev\_4.rar** formatında yüklenecektir.

2-**Son teslim tarihi 03.01.2021 pazar günü 23:55'tir.** Bu tarihten sonra ödev sistemi **otomatik kapanacak** ve **e-mail** yolu ile gönderilecek ödevler **kesinlikle dikkate alınmayacaktır.**

**3-Kopya** (doğrudan internet kaynağından alınan, arkadaştan alınarak değişken ve fonksiyon ismi değiştirilen **vb. her türlü kopya**) ödevlere **eksi puan** verilecektir.

4-Ödevler **Final notuna etki edeceği** için ödevleri yapmanız sizler için önem arz etmektedir.