

Veri Yapıları Ödev 4 (Hash)

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



Dosya	Düzen	Biçim	Gö
2214	Fatma	AYDIN	
1562	Samet	CANSIZ	
8454	Yucel	BAS	
3361	Sude	CIFTCI	
4717	Ayşe	YILMAZ	
9184	Ahmet	SURER	
5617	Ali	SURUCU	

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 hash fonksiyonu ile hesaplanan indekse atanacaktır.

Hash tablosu için boyutu 10 olan int türünde ayrı bir dizi olan hash_tablo dizisi oluşturulacaktır. Hash_tablo dizisinde müşteri numarası tutulacaktır. Hash_tablo’da başlangıçta bütün değerler (müşteri_no) “-1” olacaktır.

int hash_tablo[10];

Tabloda tutulacak değerler

hash_tablo[x] ---- > muster_i_no

Sonraki boş yerin bulunması

$(h_1(\text{müşteri_no}) + i * h_2(\text{müşteri_no})) \% \text{tablo_boyutu}$

Buradaki i iterasyon değeridir.

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. Kullanacağınız birinci hash fonksiyonu $h_1(n) = 3n + 2$ ’ dir. Burada n olarak müşteri numarası girilecek ve denklem sonucu elde edilen sonuç Hash tablosunun uzunluğu olan 10’a bölümünden kalan değer (yani $h_1(\text{Müşteri numarası}) \% 10$) olarak belirlenecektir ve hash_tablo’daki ve bağlı listedeki o indekse müşteri numarası ve musteri kaydedilecektir.

Hash tablosuna her müşteri numarası eklendiğinde, müşteri bilgileri de veri_liste’nin aynı indeksine eklenecektir.

Eğer bir çakışma olursa hash_tablo’da kayıtn girileceği yer $h_2(n) = 7n + 2$ hash fonksiyonu ile belirlenecektir.

hash_tablo	veri_liste
-1	NULL
-1	NULL
-1	NULL
-1	NULL
1234	{1234, Ahmet, SAYAN}
-1	NULL
-1	NULL
-1	NULL
-1	NULL
-1	NULL

Örneğin müşteri numarası 2214 ise hash indeksi $h_1(2214) = (2214 * 3 + 2) \Rightarrow 6644 \% 10 \Rightarrow 4$ olacaktır. $i=0$ olduğundan $i * h_2(2214) \Rightarrow 0$ olacağı için göz ardı edilmiştir. $4+0 \Rightarrow 4$. Indekse eklenecektir.

Müşteri numarası 1231 ise hash indeksi $h_1(1231) = (1231 * 3 + 2) \Rightarrow 3695 \% 10 \Rightarrow 5$ olacaktır. $i=0$ olduğundan $i * h_2(1231) \Rightarrow 0$ olacağı için göz ardı edilmiştir. $5+0 \Rightarrow 5$. Indekse eklenecektir.

Müşteri numarası 1128 ise hash indeksi $h_1(1128) = (1128 * 3 + 2) \Rightarrow 3386 \% 10 \Rightarrow 6$ olacaktır. $i=0$ olduğundan $i * h_2(1128) \Rightarrow 0$ olacağı için göz ardı edilmiştir. $6+0 \Rightarrow 6$. Indekse eklenecektir.

Müşteri numarası 1001 ise hash indeksi $h_1(1001) = (1001 * 3 + 2) \Rightarrow 3005 \% 10 \Rightarrow 5$ olacaktır. $i=0$ olduğundan $i * h_2(1001) \Rightarrow 0$ olacağı için göz ardı edilmiştir. $5+0 \Rightarrow 5$ igili indeks daha önceden dolu olduğu için i bir atırılarak işleme devam edilecektir.

$h_1(1001) = 5$

$h_2(1001) = 7 \cdot 1001 + 2 \Rightarrow 7009 \% 10 \Rightarrow 9$ olacaktır. Bundan sonra bu değerleri kullanarak hesap yapabiliriz.

$i=1$ için

$5 + 1 \cdot 9 \Rightarrow 14 \% 10 \Rightarrow 4$ daha önceden dolu (müşteri nosu 2214'den ötürü çakışma devam ediyor) bir daha artırılacak

$i=2$ için

$5 + 2 \cdot 9 \Rightarrow 23 \% 10 \Rightarrow 3$ daha önceden boş olduğu için 3. İndekse eklenecektir.

(Hash Tablosu dolduğunda “tablo dolu” mesajı verilerek daha fazla ekleme yapılamayacaktır)

2-Arama işlemi: Hash üzerinde ekleme için kullanılan formülle arama yapan fonksiyonu yazınız.

```
Aranılan Deger: 9184
9184 Ahmet SURER
Veri 3 adimda bulundu.
```

```
Aranılan Deger: 1562
1562 Samet CANSIZ
Veri 1 adimda bulundu.
```

```
Aranılan Deger: 1563
Veri bulunamadi.
```

3-Listeleme işlemi: Oluşturulan veri_liste dizisine eklenen veriler dizideki sırasına göre ekranda düzenli bir şekilde listelenecektir.

```
8454 Yuce1 BAS
3361 Sude CIFTCI
1562 Samet CANSIZ
9184 Ahmet SURER
2214 Fatma AYDIN
5617 Ali SURUCU
4717 Ayse YILMAZ
```

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