1. 9 uzunluğunda bir hash tablosu oluşturun. h(x) = k % 9.

5, 29, 20, 0, 27 ve 18 sayılarını sırasıyla tabloya ekleyin.

1. Separate Chaining
2. Linear Probing
3. Quadratic Probing
4. Double Hashing
   1. ***(hash1(key) + i \* hash2(key)) % TABLE\_SIZE***
   2. hash1(key) = key % TABLE\_SIZE
   3. hash2(key) = PRIME - (key % PRIME)
   4. PRIME = 7
5. “KastamonuÜniversitesi” string’ini 14’lük bir hash tablosuna ekleyin. h(x) = k % 14.

(k 🡪 harfin alfabedeki sırası; örn. A=1, c=3…) (Unique karakterler)

1. Separate Chaining
2. Linear Probing
3. Quadratic Probing
4. Double Hashing
   1. ***(hash1(key) + i \* hash2(key)) % TABLE\_SIZE***
   2. hash1(key) = key % TABLE\_SIZE
   3. hash2(key) = PRIME - (key % PRIME)
   4. PRIME = 7
5. Aşağıdaki örnekten yola çıkarak hash tablosu metodlarını oluşturun. h(x)=x%10

Not: Collision durumu için Linear Probing uygulanacak

int[] A =new int[10];

int HashFunction(int key);

{

//gelen key değerinden index üretecek

}

int Search(int dizi[], int veri)

{

//eğer hash tablosunda veri varsa 1, yoksa 0 döndürecek (hata mesajı main’de)

}

int Insert(int dizi[], int veri)

{

//eğer hash tablosu ‘dolu değilse’ veriyi ekleyecek, doluysa 0 döndürecek (hata mesajı main’de)

}

int Delete(int dizi[], int veri)

{

//o indexteki veri silinecek

}