

Hashing örneği:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>

#define Max_Satir 16

typedef struct _Satir
{
    int index;
    int numara;
    struct _Satir *sonraki;
}Satir;

typedef struct _Tablo
{
    Satir satirlar[Max_Satir];
}Tablo;

Tablo hash_tablosu;

int Hash(int no)
{
    return no % Max_Satir;
}

void Initialize_Table()
{
    int i;
    for (i = 0; i < Max_Satir; ++i)
    {
        hash_tablosu.satirlar[i].index = -1;
        hash_tablosu.satirlar[i].numara=-1;
        hash_tablosu.satirlar[i].sonraki = NULL;
    }
}

void Ekle(int sayi)
{
    int hash_index;
    Satir *kayit;
    hash_index = Hash(sayi);
    printf("%d ",hash_index);

    if (hash_tablosu.satirlar[hash_index].index==-1) // Collision yoksa
    {
        hash_tablosu.satirlar[hash_index].index = hash_index;
        Hash_tablosu.satirlar[i].numara= sayi;
    }
    else // Collision oluşmuş ise
    {
        kayit= (Satir *)malloc(sizeof(Satir));
        if (kayit==NULL) {
            puts("yeterli bellek yok!");
            exit(0);
        }
        kayit->index = hash_index;
        kayit->numara=sayi;
        collision(kayit, &hash_tablosu.satirlar[hash_index]);
    }
}
```

```

void collision(Satir *Kayit, Satir *Satir_Basi)
{
    Satir *gecici;
    gecici= Satir_Basi;
    while(gecici-> sonraki!=NULL)
    {
        gecici = gecici->sonraki;
    }
    gecici->sonraki= Kayit;
    Kayit->sonraki= NULL;
}

int Ara (int sayi)
{
    int hash_index, tut, adim=0;
    Satir *Kayit;
    hash_index = Hash(sayi);
    if (hash_tablosu.satirlar[hash_index].index != -1)
    {
        if (hash_tablosu.satirlar[hash_index].numara== sayi)
            return hash_tablosu.satirlar[hash_index].index;
        else {
            Kayit = hash_tablosu.satirlar[hash_index].sonraki;
            while(Kayit!= NULL)
            {
                if (Kayit->numara==sayi) return Kayit->index;
                Kayit= Kayit->sonraki;
            }
        }
    }
    return -1; // Başarısız arama
}

void Tablo_Listele()
{
    int i;
    Satir *Kayit;
    puts("\n***Tablo***");
    for (i =0; i < Max_Satir; ++i)
    {
        if (hash_tablosu.satirlar[i].index != -1)
        {
            printf("%2d:%s\n",hash_tablosu.satirlar[i].index,
hash_tablosu.satirlar[i].numara);
            Kayit = hash_tablosu.satirlar[i].sonraki;
            while(Kayit!=NULL)
            {
                printf("%2d: %s ", Kayit->index, Kayit->numara);
                Kayit= Kayit->sonraki;
            }
        }
    }
}

int main()
{
    int i,index;
    int numaralar[Max_Satir]= {15,26,6,58,35,86,21,14,18,27,33,10,7,16};
    Initialize_Table();
    for (i = 0; i < Max_Satir; ++i) Ekle(numaralar[i]);
    int aranan=numaralar[5];
    index= Ara(aranan);
    printf ("\n %d numaralı kayıt %d. indekstedir", aranan, index)
    Tablo_Listele();
    getch();
    return 0;
}

```