

Tek List.cpp

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

typedef struct Liste
{
    int numara;
    struct Liste *sonraki;
}BListe;

BListe *ilk=NULL,*son=NULL;

BListe * Ara(int num)
{
    BListe *gecici;
    gecici=ilk;
    while(gecici)
    {
        if(gecici->numara==num) return gecici;
        gecici=gecici->sonraki;
    }
    return NULL;
}
```

```
int Listele()
{
    BListe *gecici;
    gecici=ilk;
    if (gecici==NULL) return 0;
    while(gecici)
    {
        printf("%d\n",gecici->numara);
        gecici=gecici->sonraki;
    }
    return 1;
}
```

```

void Silme(int silnum)
{ BListe *gecici,*bironceki;
  gecici=ilk;
  bironceki=NULL;
  while(gecici)//silinecek kayıt konumunu bul
  {
    if (gecici->numara==silnum) break;
    bironceki=gecici;
    gecici=gecici->sonraki;
  }
  if (gecici!=NULL)//silinecek kayıt yoksa geçici NULL'u gösteriyordur
  {
    if (gecici==ilk)//silinecek ilk kayıtsa
    { if (ilk==son)//ilk ve tek kayıt var ise
      {
        ilk=NULL;
        son=NULL;
      }
      else //ilk kayıt ise ancak birden fazla kayıt var ise
      {
        ilk=ilk->sonraki;
      }
    }
    else //silinecek kayıt ilk kayıt değil ise
    if (gecici==son)//silinecek kayıt son kayıt ise
    {bironceki->sonraki=NULL;
     son=bironceki;
    }else //silinecek kayıt arada bir kayıt ise
    {bironceki->sonraki=gecici->sonraki;}

    free(gecici);
  }
}

```

Örnek: numara bilgisinin tutulacağı çift yönlü bir bağlı liste veri yapısı tanımlayınız. Listeye

- a) Ekleme
- b) Tüm kayıtları Listeleme
- c) Arama
- d) Silme

Fonksiyonlarını yazınız.

Çift Yönlü Bağlı Liste

Cift_lis.cpp

```
typedef struct Liste
{
    int numara;
    struct Liste *sonraki;
    struct Liste *onceki;
}BListe;
BListe *ilk=NULL,*son=NULL;

BListe *Bilgi_Al(int num)
{
    BListe *bilgi=(BListe *)malloc(sizeof(BListe));
    Bilgi->numara=num;
    Bilgi->sonraki=NULL;
    Bilgi->onceki=NULL;
    return bilgi;
}

void Ekle(BListe *bilgi)
{
    if (ilk==NULL)
    {
        ilk=bilgi;
        son=ilk;
    }
    else
    {
        son->sonraki=bilgi;
        bilgi->onceki=son;
        son=bilgi;
    }
}
```

