

Çift Yönlü Bağlı Liste

Cift_lis.cpp

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

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

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

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=Ara(silnum);
  if (geçici==NULL) Printf("Silinmek istenen kayıt listede yoktur");
  else // Silinmek istenen kayıt listede varsa
  {
    if (ilk==gecici)//silinecek ilk kayıtsa
    { if (ilk->sonraki==NULL)//ilk ve tek kayıt var ise
      {
        ilk=NULL;
        son=NULL;
      }
      else //ilk kayıt ise ancak birden fazla kayıt var ise
      {
        ilk=ilk->sonraki;
        ilk->onceki=NULL;
      }
    }
    else //silinecek kayıt ilk kayıt değil ise
    {
      if(gecici==son)//silinecek son kayıt ise
      {son=son->onceki;//son kayıt ekleme için kaybedilmemeli.
        son->sonraki=NULL;
      }
      else //silinecek aradaki kayıt arada ise
      {
        gecici->onceki->sonraki=gecici->sonraki;
        gecici->sonraki->onceki=gecici->onceki;
      }
    }
    free(gecici);
  }
}

```