

BLM-112 PROGRAMLAMA DİLLERİ II

Ders-7 Tek Bağlı Doğrusal Listeler

Yrd. Doç. Dr. Ümit ATILA

umitatila@karabuk.edu.tr

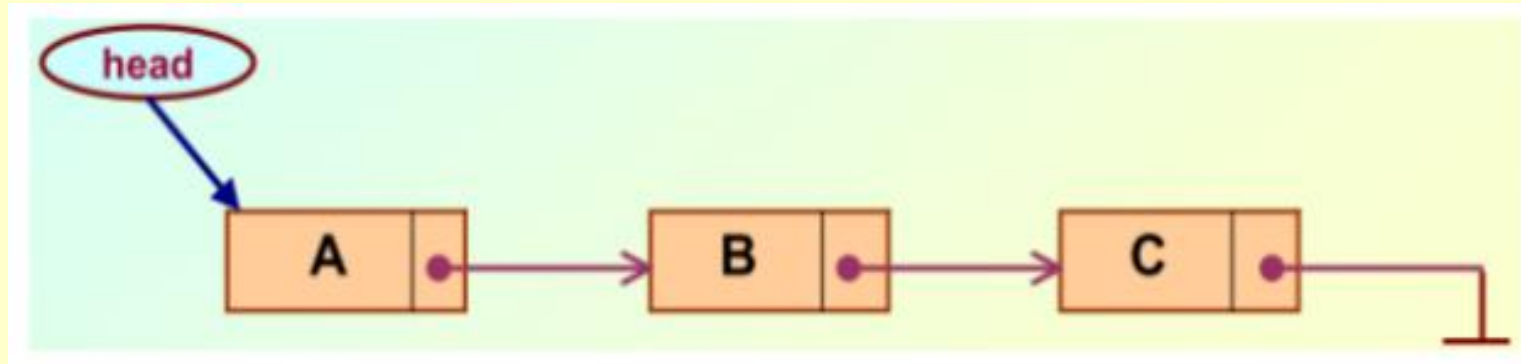
<http://web.karabuk.edu.tr/umitatilla/>

BAĞLI LİSTELER

- Bağlı listeler konusuna çalışmanın iki yönden faydası var.
 - Bağlı listeler gerçek programlarda kullanılabilecek bir veri yapısıdır. Bağlı listelerin güçlü ve zayıf yönlerini bilmek algoritmaların çalışma zamanı karmaşıklığı, kapladığı alan karmaşıklığı yönlerini düşünmenize yardımcı olur.
 - Bağlı listeler pointer'ların anlaşılması için iyi bir yöntemdir.

BAĞLI LİSTELER

- Bağlı liste çalışma zamanı sırasında değişebilen bir veri yapısıdır.
 - Ardışık elemanlar pointer ile bir birine bağlanır.
 - Son eleman NULL değeri gösterir.
 - Programın çalışması sırasında boyutu büyüyebilir veya küçülebilir.

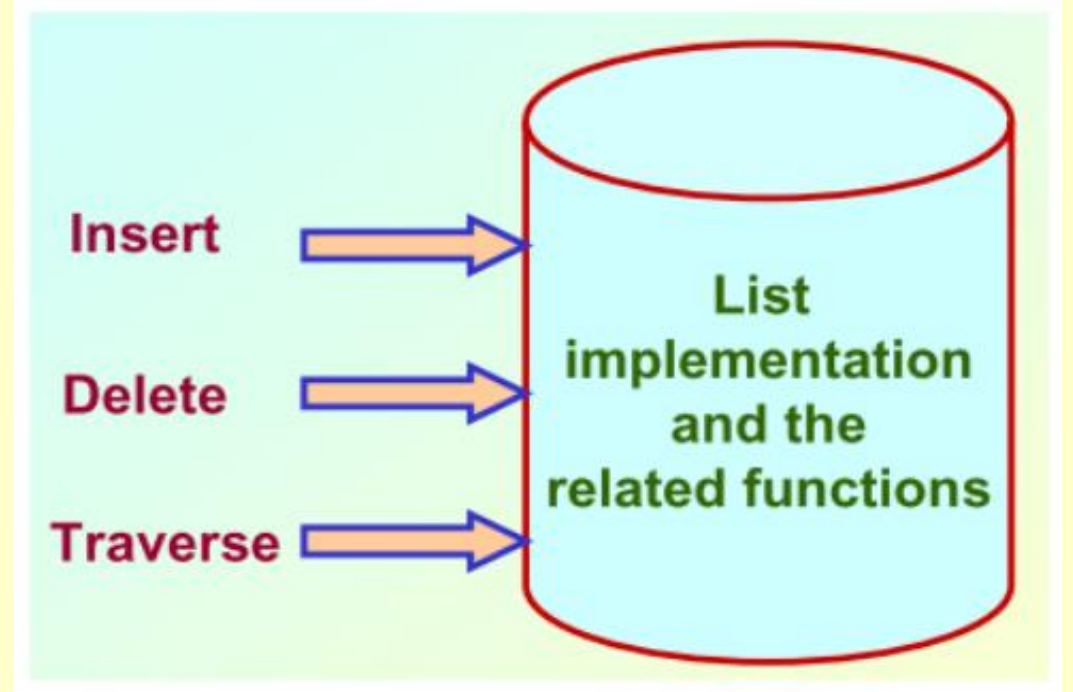


DİZİLER vs. BAĞLI LİSTELER

- Diziler aşağıdaki hususlarda uygundur:
 - En sona eleman ekleme ve en sondan eleman silme
 - Her hangi bir elemana erişme
- Bağlı listeler aşağıdaki hususlarda uygundur:
 - Eleman ekleme
 - Eleman silme
 - Sıralı erişim gerektiren uygulamalar
 - Eleman sayısının önceden tahmin edilemediği durumlar

Bağlı Listeler

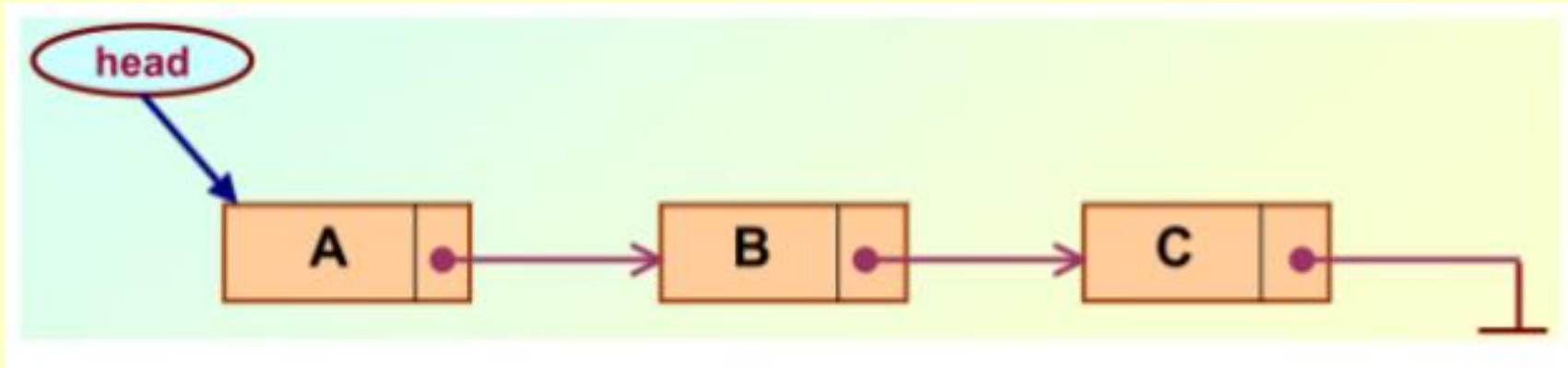
- Bağlı listelerin veri tipi programcı tarafından tasarlanır.
- int, float gibi veri tiplerinden daha karmaşık olur.
- Listelerde genel amaç ;



Bağlı Listelerde Temel İşlemler

- Liste oluşturma
- Listede dolaşma
- Listeye eleman ekleme
- Listeden eleman silme
- İki listeyi birleştirme

Tek Bağlı Doğrusal Liste



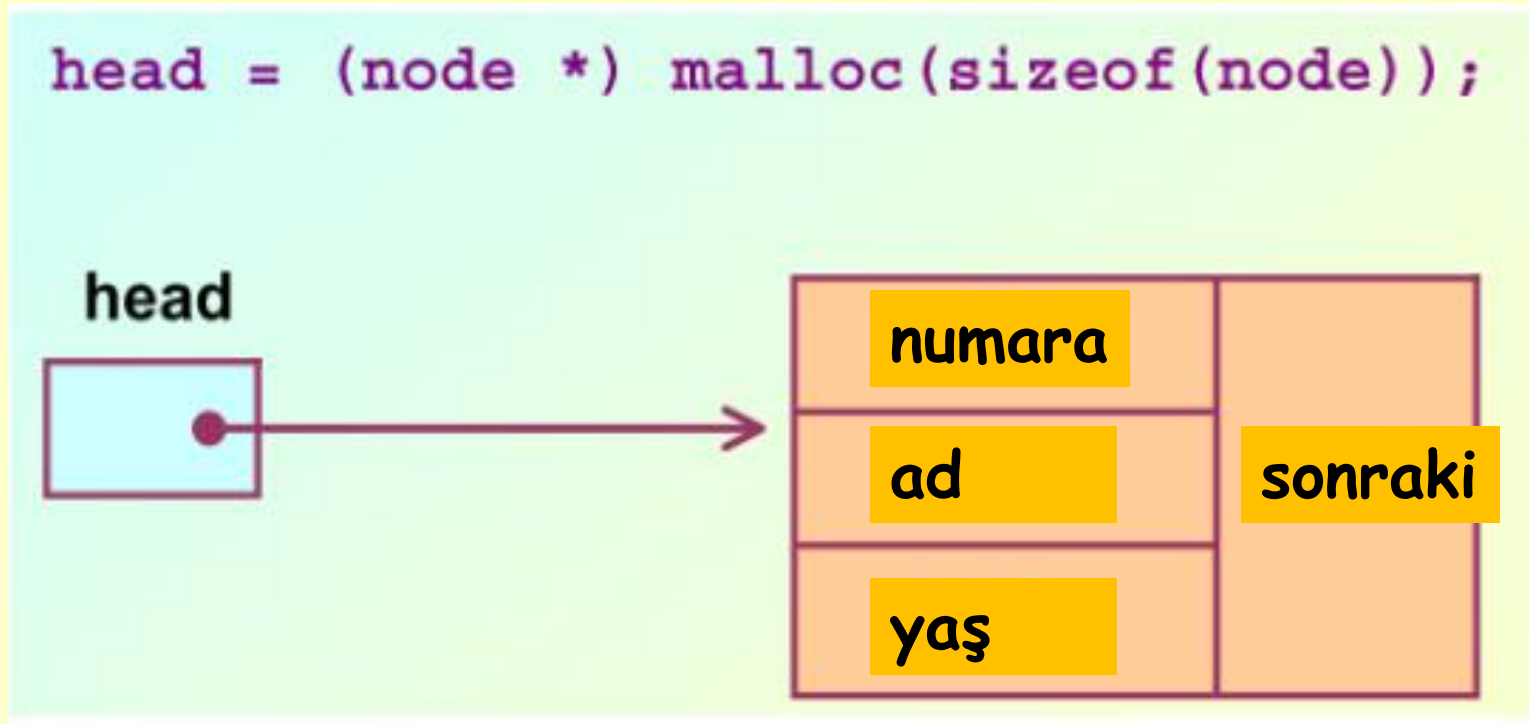
Liste Düğüm Yapısı

- Listedeki bir düğümün yapısının şu şekilde olduğunu düşünelim:

```
struct ogrenci{  
    int no;  
    char ad[40];  
    int vize;  
    struct ogrenci *sonraki;  
};  
typedef struct ogrenci ogrenci;
```

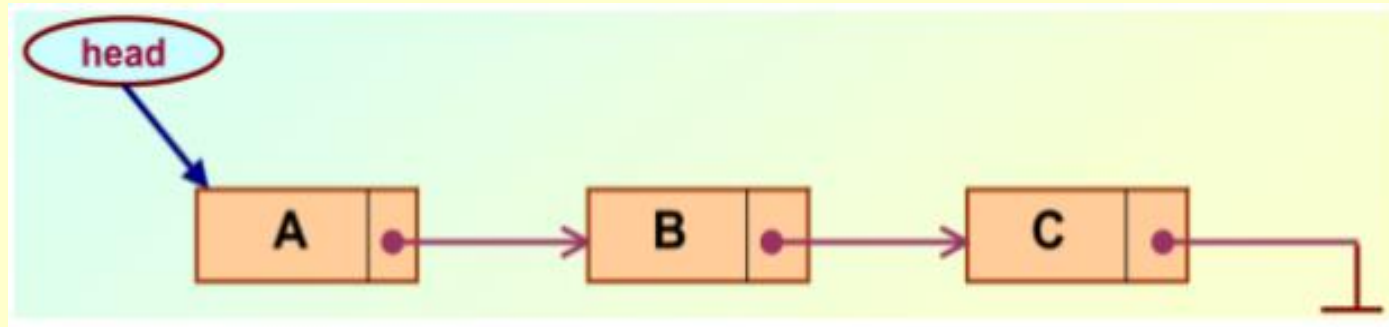

Doğrusal Liste Oluşturma

- İlk önce bir düğüm oluşturulmalı ve head'in bu düğümü göstermesi sağlanmalıdır.



Doğrusal Liste Oluşturma

- Eğer başlangıç bağlı listesinde N adet düğüm olacaksa:
 - N adet kayıt için teker teker hafızadan yer ayır.
 - Kayıtların alan bilgileri girilir.
 - Kayıtların bağlantıları düzenlenir böylece zincir kurulur.



Doğrusal Liste Oluşturma

```
ogrenci * listeOlustur()
{
    ogrenci *p, *h;
    int i, n;
    printf("Kac dugum:");
    scanf("%d", &n);
    for(i=0;i<n;i++)
    {
        if(i==0)
        {
            h = (ogrenci *) malloc(sizeof(ogrenci));
            p = h;
        }
        else
        {
            p->sonraki = (ogrenci *) malloc(sizeof(ogrenci));
            p = p->sonraki;
        }
        printf("No gir:");
        scanf("%d", &p->no);
        printf("Ad gir:");
        scanf("%s", p->ad);
        printf("Vize gir:");
        scanf("%d", &p->vize);
    }
    p->sonraki = NULL;
    return h;
}
```

Doğrusal Listede Dolaşma

- Liste oluşturulduktan sonra ve head düğümü listenin ilk düğümünü gösterir hale gelince:
 - Pointer'ları takip et.
 - Düğümlere sıra geldikçe onların içeriklerini yazdır.
 - Sıradaki pointer NULL değeri gösteriyorsa dur.

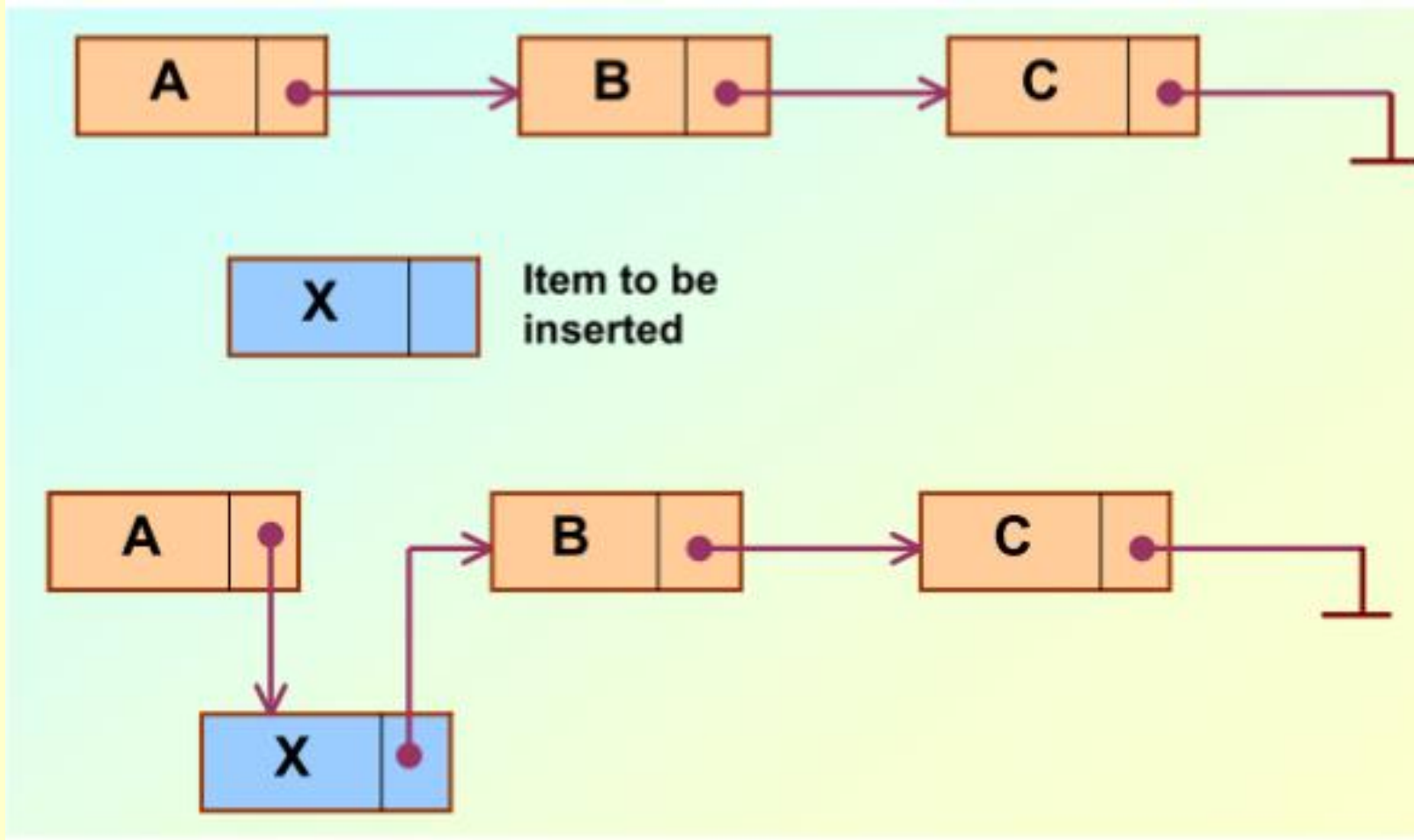
Doğrusal Listede Dolaşma

```
void listeDolas(ogrenci *h)
{
    ogrenci *p;
    p = h;
    while(p!= NULL)
    {
        printf("*****\n");
        printf("No: %d \t\t Ad: %s\n", p->no, p->ad);
        printf("Vize: %d\n", p->vize);
        p = p->sonraki;
    }
}
```

Listeye Düğüm Ekleme

- Ekleme için:
 - Yeni kayıt oluşturulur.
 - Yeni kaydın sonraki göstericisi kendinden bir sonra gelecek kaydı gösterecek şekilde ayarlanır.
 - Yeni kayıttan önceki kaydın sonraki göstericisi yeni kaydı gösterecek şekilde ayarlanır.
- Belli bir düğümden önce düğüm ekleme biraz daha karışıktır.
 - Burada anahtar olarak isimlendirilen bir değer kullanılır.
 - Bizim örneğimizde anahtar alan «numara».

Listeye Düğüm Ekleme



Listeye Düğüm Ekleme

- Düğüm başa ekleniyorsa
 - Sadece bir tane «sonraki» göstericisi değiştirilir.
 - Head, yeni düğümü gösterecek şekilde ayarlanır.
 - Yeni düğüm daha önce ilk düğüm olan düğümü gösterir.
- Düğüm sona ekleniyorsa
 - İki tane <sonraki» göstericisi değiştirilir.
 - Son düğüm artık yeni düğümü gösterir.
 - Yeni düğüm NULL gösterir.
- Düğüm araya ekleniyorsa
 - İki tane <sonraki» göstericisi değiştirilir.
 - Yeni düğüm önceki düğümün sonrakini gösterir.
 - Önceki düğüm artık yeni düğümü gösterir.

Listeye Düğüm Ekleme

```
ogrenci * dugumEkle(ogrenci *h)
{
    int kayitNo;
    ogrenci *p, *q, *yeni;
    p = h;

    printf("HANGI kayittan once eklensin:");
    scanf("%d", &kayitNo);

    yeni = (ogrenci *) malloc(sizeof(ogrenci));
    printf("No gir:");
    scanf("%d", &yeni->no);
    printf("Ad gir:");
    scanf("%s", yeni->ad);
    printf("Vize gir:");
    scanf("%d", &yeni->vize);

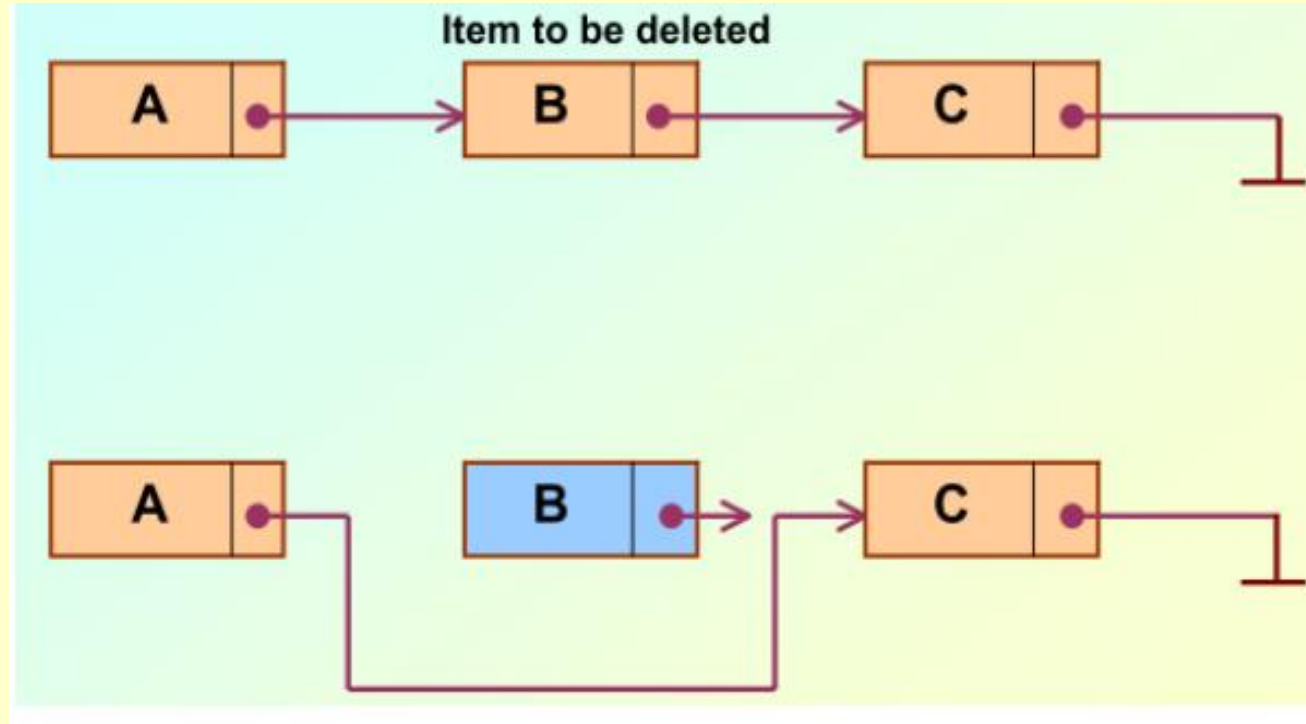
    if(p->no == kayitNo) // Başa ekleme
    {
        yeni->sonraki = p;
        h = yeni;
    }
```

```
else
{
    while(p->no != kayitNo && p->sonraki != NULL)
    {
        q = p;
        p = p->sonraki;
    }

    if(p->no == kayitNo) //araya ekleme
    {
        q->sonraki = yeni;
        yeni ->sonraki = p;
    }
    else if(p->sonraki == NULL) //sona ekleme
    {
        p->sonraki = yeni;
        yeni->sonraki = NULL;
    }
}
return h;
}
```

Listeden Dügüm Silme

- Silinecek düğümden bir önceki düğümün sonraki göstericisi değiştirilerek silinecek düğümden sonraki düğümü göstermesi sağlanır.



Listeden Düğüm Silme

- Belli bir düğümü silmek için (numara bilgisi verilen düğüm) 2 durum söz konusudur:
 - Silinecek düğüm ilk düğüm olabilir.
 - Silinecek ilk düğümün haricindeki diğer bir düğüm olabilir.

Listeden Düğüm Silme

```
ogrenci * dugumSil(ogrenci *h)
{
    int kayitNo;
    ogrenci *p, *q;
    p = h;

    printf("Silmek istedigim kayit no gir:");
    scanf("%d", &kayitNo);
    if(p->no == kayitNo) //baştaki düğüm silme
    {
        h = p->sonraki;
        free(p);
    }
}
```

```
else
{
    while(p->no != kayitNo && p->sonraki != NULL)
    {
        q = p;
        p = p->sonraki;
    }

    if(p->no == kayitNo)
    {
        q->sonraki = p->sonraki;
        free(p);
    }
    else if(p->sonraki == NULL)
    {
        printf("Silinecek dugum bulunamadi");
    }
}
return h;
}
```

Tek Bağlı Doğrusal Liste Uygulaması

```
void main()
{
    ogrenci *head;
    int s;
    while(1)
    {
        printf("1-Liste Olustur\n"
              "2-Liste Dolas\n"
              "3-Dugum EKle\n"
              "4-Dugum Sil\n"
              "5-Çikis\n");
        printf("Secim yap:");
        scanf("%d", &s);
        if(s == 1)
            head = listeOlustur();
        else if(s == 2)
            listeDolas(head);
        else if(s == 3)
            head = dugumEkle(head);
        else if(s == 4)
            head = dugumSil(head);
        else if(s == 5)
            exit(0);
        else
            printf("Yanlis secim yaptiniz...\n");
    }
}
```

Kaynaklar

- Paul J. Deitel, "C How to Program", Harvey Deitel.
- Kaan Aslan, "A'dan Z'ye C Klavuzu 8. Basım", Pusula Yayıncılık, 2002.