

Kuyruk işlemleri 2

```
//Tekli Linked list ilk elemanı için kullanılacak değişkene boş değer atıyoruz.
static Tekli head = null;

//Tekli Liste içindeki ilk eleman alınıyor.
//Tekli liste ilk elemanı listenin sonraki elemanı olarak atanıyor.
static int elamanaltekli()
{
    int data = head.data;
    head = head.next;
    return data;
}

//Tekli Liste içine kayıt ekleniyor.
static void ekletekli(int data)
{
    Tekli q = new Tekli();
    q.data = data;
    q.next = null;
    if (head == null)
    {
        //Linked list içinde hiç kayıt yoksa head değeri yeni eklenen oluyor.
        head = q;
    }
    else
    {
        //Listede kayıt varsa listenin son elemanı ilk elemanı olarak gösteriliyor.
        //Liste içindeki next değeri boş oluncaya kadar yani son elemana kadar dönülüyor.
        Tekli last = head;
        while (last.next != null)
        {
            //Her Last değeri Listenin sonraki kaydı olarak güncelleniyor.
            last = last.next;
        }
        //döngü sonucu bulunan Tekli Listenin son kaydının nexti yeni eklenen q değeri olarak ekleniyor.
        last.next = q;
    }
}
```

```
//Çiftli Linked Listin ilk ve son elemanları için kullanılacak değişkenlere
boş değer atıyoruz
static Ciftli rearCiftli = null;
static Ciftli frontCiftli = null;

//Çiftli Linked List İçin Kuyruğa ekleme işlemi yapıyoruz.
static void ekleCiftli(int data)
{
    Ciftli q = new Ciftli();
    q.data = data;
    q.next = null;
    q.prev = null;
    if(frontCiftli == null)
    {
        //Çiftli Linked Listede hiç eleman olmadığı için Kuyruğun ilk ve son
        Kaydı yeni oluşturulan kayıt
        frontCiftli = q;
        rearCiftli = q;
    }
    else
    {
        //son tanımlanan rearCiftli verisinin sonraki kaydı yeni eklenen değer
        olarak belirleniyor
        rearCiftli.next = q;
        //yeni eklenen kaydın önceki değeri Mevcut rearCiftli kaydı oluyor
        q.prev = rearCiftli;
        //Son olarak son kayıt olacak rearCiftli verisi yeni eklenen kayıt
        yapılıyor.
        rearCiftli = q;
    }
}

//Çiftli Linked List içinde Eleman Alınıyor
static int elamanalCiftli()
{
    //Çiftli Linked list içindeki ilk kayıt datası alınıyor
    int data = frontCiftli.data;
    //Çiftli linked listin ilk kaydının sonraki değeri kontrol ediliyor.
    if (frontCiftli.next != null)
    {
        //Çiftli Linked listin ilk elemanı sonraki kaydının önceki verisi boş
        //burada cümle saçma gibi geliyor ama yapılan işlem bu. böylece bu kayıt
        içinde veri olmayacak
        frontCiftli.next.prev = null;
    }
    //Çiftli Linked Listin ilk kaydı ilk kayıttan sonra gelen kayıt olarak
    atanıyor.
    frontCiftli = frontCiftli.next;
    return data;
}
```

```
//Çiftli Liste Sınıfı
public class Ciftli
{
    public Ciftli next;
    public Ciftli prev;
    public int data;
}

//Tekli Liste Sınıfı
public class Tekli
{
    public Tekli next;
    public int data;
}

static void Main(string[] args)
{
    //Tekli Linked listenin içine 20 eleman dolduruyoruz
    for (int i = 0; i < 20; i++)
    {
        ekletekli(i);
    }

    //Tekli Linked List içine Doldurulan bu 20 elemanı ekrana yazdırıyoruz.
    for (int i = 0; i < 20; i++)
    {
        Console.WriteLine(elamanaltekli());
    }

    //Çiftli Linked listenin içine 20 eleman dolduruyoruz
    for (int i = 0; i < 20; i++)
    {
        ekleCiftli(i);
    }

    //Tekli Linked List içine Doldurulan bu 20 elemanı ekrana yazdırıyoruz.
    for (int i = 0; i < 20; i++)
    {
        Console.WriteLine(elamanalCiftli());
    }

    Console.ReadLine();
}
```