

KUYRUK YAPISI

Kuyruk yapısı ilk giren ilk çıkar mantığıyla çalışan bir yapıdır. (Yazıcı kuyruğu gibi)

1) Dizi üzerinde kaydırmalı Kuyruk Yapısı

Son pozisyona eklenir

1	8	4	7	
---	---	---	---	--

İlk pozisyonundan çıkar

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

#define KBoyut 100
typedef struct Kuyruk
{
    int son;
    int eleman[KBoyut];
}Kuyruklar;

Kuyruklar Yeni_Kuyruk;
Yeni_Kuyruk.son=-1;

int Kuyruk_Dolumu()
{
    if (Yeni_Kuyruk.son>=KBoyut-1) return -1;else return 1;
}

int Kuyruk_Bosmu()
{
    if (Yeni_Kuyruk.son==-1) return -1;else return 1;
}

void Kuyruuga_Ekle(int ekle)
{
    if (Kuyruk_Dolumu(Yeni_Kuyruk)==-1)
    {
        printf("Kuyruk Dolu\n");
    }
    else
    {
        Yeni_Kuyruk.son++;
        Yeni_Kuyruk.eleman[Yeni_Kuyruk.son]=ekle;
    }
}
```

```

int Kuyruktan_Cikar()
{int cikan_eleman,i;
  if (Kuyruk_Bosmu(Yeni_Kuyruk)==-1)
  { printf("Kuyruk Bos\n");
    return -1;
  }
  else
  {
    cikan_eleman=Yeni_Kuyruk.eleman[0];
    for(i=1;i<=Yeni_Kuyruk.son;i++)
      Yeni_Kuyruk.eleman[i-1]=Yeni_Kuyruk.eleman[i];
    Yeni_Kuyruk.son--;
    return cikan_eleman;
  }
}

```

~~10~~ 20 30 40
 0 1 2 3

```

void Listele(Kuyruklar *Yeni_Kuyruk)
{int i;
  for (i=0;i<=Yeni_Kuyruk.son;i++)
    printf("\n: %d",Yeni_Kuyruk.eleman[i]);
}

Kuyruklar Yeni_Kuyruk;
Yeni_Kuyruk.son=-1;

```

```

void main()

{
    int i;
    char secim;
    int numara;
    clrscr();
    while(1==1)
    {
        clrscr();
        puts("\nEkleme\nCikarma\nListeleme\nCikis\nSecim?");
        secim=getchar();
        switch(secim)

        {
            case 'e':
                puts("Numarayi giriniz");
                scanf("%d",&numara);
                Kuyrugan_Ekle(numara);
                break;
            case 's':i=Kuyruktan_Cikar();
                printf("Cikan=%d",i);
                break;
            case 'l':
                Listele(Yeni_Kuyruk);
                getch();
                break;
            case 'c': exit(0);

        }
    }
}

```