

2. YIĞIN YAPISI

Yığınlar ilk giren son çıkar (last in/first out) prensibine göre elaman erişiminin sınırlandığı bir veri yapısıdır. Ekleme 1,2,3,4->4,3,2,1 şeklinde alınır.

4
3
2
1

Örnek : (Dizi ile yığın tanımlama)

```
#define N 100
int Yigin[N], indis=-1;
int Yigina_Ekle(int veri)
{
    if (indis>=N-1)
        {puts("Yigin Dolu");return -1;}
    else
    {
        indis++;
        Yigin[indis]=veri;
    }
}
int Yigindan_Al()
{int cikan;
    if (indis== -1)
    {
        puts("Yigin Bos");
        return -1;
    }else
        cikan=Yigin[indis];
        indis--;
        Return cikan;
}
```

Örnek : Tam sayı elemanların tutulacağı ve dizi veri yapısı üzerinden bir yığın oluşturunuz.

- Yığının dolu olup olmadığını kontrol ederek, yığına ekleme işlemi gerçekleştiren yığın dolu ise “Yığın Dolu” mesajı veren bir fonksiyon yazınız.
- Yığının boş olup olmadığını kontrol ederek, yığından çıkarma işlemi gerçekleştiren yığın boş ise “Yığın Boş Çıkarma Yapılamaz” mesajı veren bir fonksiyon yazınız.
- Yığındaki tüm elemanları yığından çıkış sırasına göre listeleyen bir Listeleme fonksiyonu yazınız.

“Yigin.cpp”

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

```
#define YBoyut 100
typedef struct Yigin
{
    int indis;
    int eleman[YBoyut];
}Yiginlar;
```

```
Yiginlar Yeni_Yigin;
Yeni_Yigin.indis=-1;
```

```
int Yigin_Dolumu()
{
    if (Yeni_Yigin.indis>=YBoyut-1) return -1;else return 1;
}
```

```
int Yigin_Bosmu()
{
    if (Yeni_Yigin.indis==-1) return -1;else return 1;
}
```

```
void Yigina_Ekle(int ekle)
{
    if (Yigin_Dolumu(Yeni_Yigin)==-1)
    {
        printf("Yigin Dolu\n");
    }
    else
    {
        Yeni_Yigin.indis++;
        Yeni_Yigin.eleman[Yeni_Yigin->indis]=ekle;
    }
}
```

```

int Yigindan_Cikar()
{int cikan_eleman;
    if (Yigin_Bosmu(Yeni_Yigin)==-1)
        { printf("Yigin Bos\n");
          return -1;
        }
    else
        {
            cikan_eleman=Yeni_Yigin->eleman[Yeni_Yigin->indis];
            Yeni_Yigin.indis--;
            return cikan_eleman;
        }
}

```

```

void Listele()
{int i;
    for (i=Yeni_Yigin.indis;i>=0;i--)
        printf("\n %d",Yeni_Yigin->eleman[i]);
}

```

```

void main()
{
    char secim;
    int numara;
    clrscr();
    while(1==1)
    {
        clrscr();
        puts("\nEkleme\nCikarma\nListeleme\nCikis\nSecim?");
        secim=getchar();
        switch(secim)
        {
            case 'e':

```

```

        puts("Numarayı giriniz");
        scanf("%d",&numara);
        Yigina_Ekle(numara);
        break;
    case 's':    numara=Yigindan_Cikar();
                printf("Yigindan Cikan=%d",numara);
                break;
    case 'l':
                Listele();
                getch();
                break;
    case 'c':exit(0);

    }
}
}

```

Örnek : (Desimal den Binary ‘e dönüşüm)

“Yigin_bi.cpp”

10 tabanındaki bir sayıyı 2’lik tabana yığın veri yapısı kullanarak dönüştürecek fonksiyonu yazınız. *(Tüm yığın fonksiyonlarını yeniden yazmadan yukarıda yazdığımız fonksiyonları hazır olarak kullanalım).* Yukarıdaki tamsayı tipi tutan yığın yapısına eklenen fonksiyonlar.

```

void cevir(int sayi)
{ int digit;
  while(sayi>0)
  {
      digit=sayi%2;
      Yigina_Ekle(digit);
      sayi=sayi/2;
  }
  while (Yeni_Yigin.indis>=0)
  {
      digit=Yigindan_Cikar();
      printf("%d",digit);
  }
}

Yiginlar Yeni_Yigin;
Yeni_Yigin.indis=-1;

void main()
{
    int numara;
    clrscr();
    puts("\n 2lige cevrilecek Sayi?");
    scanf("%d",&numara);
    cevir(numara);
}

```