

## RECURSION

Kendi kendini çağıran programlardır. Özyinelemeli bir programın kendi kendini çağırması dışında bir de durabilmesi için özyinelemeli olmayan bitme koşuluna (termination condition) sahip olması gerekir.

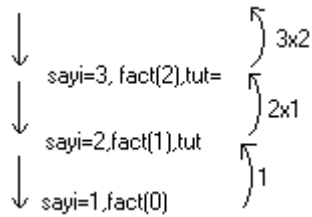
**Örnek:** Faktöryel örneği

1 if n=0;  
 $n \times (Faktoryel(n-1))$  if n!=0

Bu tanımlamada 0 bitme koşuludur.

```
double factr(int sayi) // recursive faktöriyel hesabı
{
    if (sayi==0) return 1;
    else
    {
        tut=factr(sayi-1);
        return sayi*tut;
    }
}
```

veya  
return sayi\*fact(sayi-1);



```
double fact(int sayi) // döngüsel faktöriyel hesabı
{int i;
    double factN=1;
    for (i=1;i<=sayi;i++)
        factN=factN*i;
    return factN;
}
```

```
void main()
{int sayi;
double tut;
clrscr();
printf("sayi\n");
scanf("%d", &sayi);
printf("Sonuc=%0.0f", fact(sayi));
}
```

**Örnek:** Fibonacci sayılar. 0,1,1,2,3,5,8,13,21,...

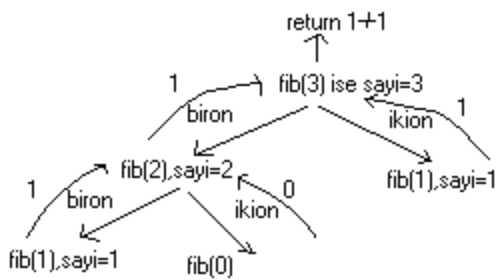
Serisi fibonacci serisi olarak adlandırılır.

$$\text{Fib} = \begin{cases} n, & \text{eğer } n \leq 1 \\ \text{fib}(n-1) + \text{fib}(n-2), & \text{eğer } n > 1 \end{cases}$$

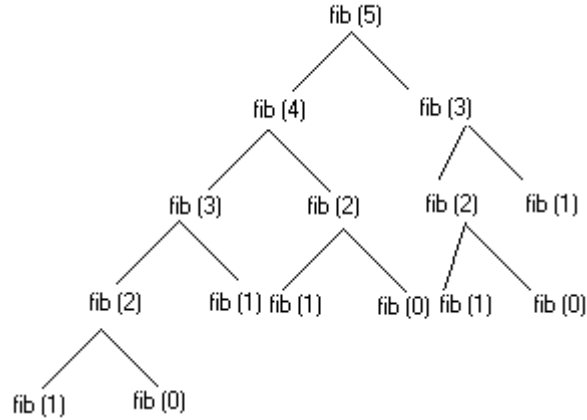
```
void fib(int sayi) // Döngüsel fibonacci serisi hesabı
{
    int i, sonfib=1, onfib=0, tut=0;
    printf("%d %d", onfib, sonfib);
    for (i=0; i<=sayi; i++)
    {
        tut=sonfib+onfib;
        onfib=sonfib;
        sonfib=tut;
        printf(" %d", tut);
    }
}
```

```
int fibr (int n) // recursive fibonacci serisi hesabı
{
    int biron, ikion;
    if (N<=1) return n;
    biron=fibr(n-1);
    ikion=fibr(n-2);
    return biron+ikion;
}
```

Bu örnek özyineleme kavramının gücünü göstermekten öte, özyinelemenin bazı durumlarda ne denli kötü çalışabileceğini gösteren bir örnektir. Fibonacci sayısı için verilen  $N$  değeri için özyinelemesiz olarak yaklaşık  $N$  adet işlemin (döngü) yapılmasına karşılık, özyinelemeli algoritmada örneğin  $\text{fib}(2)$  için 3,  $\text{fib}(3)$  için ise 2 defa tekrar hesaplama yapılmaktadır. Aşağıdaki örnek **fib(3)** ün hesaplanması göstermektedir.



Aşağıdaki ifade **fib(5)**'in hesaplanmasını göstermektedir.



**Örnek:** Binary arama algoritmasının özyinelemeli olarak gerçekleştirilmesi.

```

int bin_search(int dizi[],int bas,int son,int aranan)
{ int orta,orta_deger;
  if (bas>son) return -1;  // Başarısız arama
  else
  {
    orta=(bas+son)/2;
    orta_deger=dizi[orta];
    if(aranan==orta_deger) return orta;    // Başarılı arama
    else
    if (aranan<orta_deger)
      return bin_search(dizi,bas,orta-1,aranan); // sol parça
    else
      return bin_search(dizi,orta+1,son,aranan); // sağ parça
  }
}

```

**Örnek:** tekyönlü bir bağlı liste elemanlarını tersten ekrana yazdıracak bir özyinelemeli ters\_yaz() fonksiyonu yazınız. Bağlı listede kayıtlar mevcut kabul edilecek, oluşturulmayacaktır. Listenin ilk elemanı \*ilk değişkeninde, son elemanı NULL ile belirlendiğini kabul ediniz.

```

void Ters_Yaz(Bliste *gecici)
{
  if(gecici->sonraki!=NULL) Ters_Yaz(gecici->sonraki);
  printf("%d",gecici->numara);
}

```