



## Özyineleme / tekrarlamalı (Recursive / Recursion)

Kendini doğrudan veya dolaylı olarak çağıran fonksiyonlara **özyineli** (recursive) fonksiyonlar adı verilir.

Fonksiyon içersinden fonksiyon çağırmanın özel bir hâlidir.

Özyineleme;

şartı sağladığı sürece bir başka fonksiyon yerine kendisini çağırmaya devam ederek tekrarlı işlemlerin çözümüne farklı bir bakı açısı getirir.

- bir problemi çözmek için problemi iki veya daha fazla alt probleme bölen bir yöntemdir.
- iterasyonun (döngüler) yerine geçebilecek güçlü bir programlama tekniğidir.
- daha yalın bir kod imkanı sunar

## Özyineleme (Recursion) VS İterasyon (Iteration)

- Herhangi bir fonksiyonun iteratif yani tekrarlı versiyonu, özyineli versiyonundan zaman (time) ve yer (space) bakımından genelde daha etkindir. Bunun nedeni, özyinelemede fonksiyonun her çağrılığında fonksiyona giriş ve çıkışta oluşan yüklerdir. aynı fonksiyonu çağırdığında eski fonksiyonun stack yapısı bozulmaz ve derinlere gittikçe stack artar.
- Genelde yapısı uygun olan problemlerin çözümünde özyinelemenin kullanılması daha doğal ve mantıklıdır.
- Tanımlamalardan çözüme doğrudan ulaşılabilir.
- İterasyonda "Control Structure" olarak döngüler yolu ile tekrar kullanılırken, özyinelemede seçim yapısı kullanılır.
- İterasyon, döngü durum şartı geçersizliğinde sonlanırken, özyineleme en basit duruma (simplest case = base case) ulaşıldığında sonlanır.
- İterasyonda kullanılan savaç değeri değiştirilerek problem çözülürken, özyinelemede orijinal problemin daha basit sürümleri oluşturularak çözüme ulaşılır.

## Özyineleme - örnek

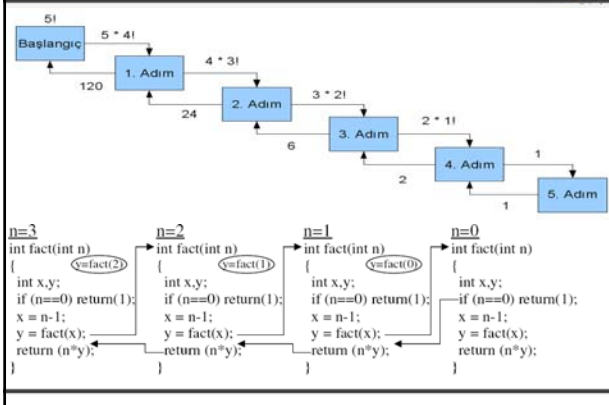
Faktöriyel Hesabı ( $n! = n \cdot (n-1)!$ )

```
#include <iostream.h>
```

```
long faktoriyel(int x) {
    if(x<=1)
        return 1; // dönüş değeri
    else
        return (x*faktoriyel(x-1));
    // özyineleme
}
```

```
long faktoriyeliterative(int n) {
    int x, f=1;
    for(x=n; x>0; x--)
        f *= x;
    return (f);
}
```

## Özyineleme - örnek



## Özyineleme - örnek

```
#include <iostream.h>
```

```
int f(int k) {
```

```
    if (k==1) return 1;
    if (k==2) return 3;
    return f(k-1) + k;
}
```

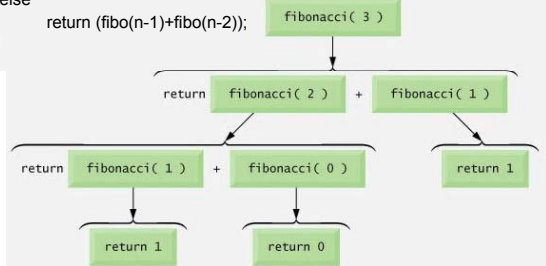
```
main() {
```

```
    cout<<"f="<<f(10);
    system("pause");
}
```

## Özyineleme - örnek

Fibonacci 1 1 2 3 5 8 13 21 34 45 ..... Fibo(n)= Fibo(n-2)+Fibo(n-1)

```
long fibo(int n){
    if(n==2 || n==1) return 1;
    else
        return (fibo(n-1)+fibo(n-2));
}
```



Dr. Şenar Turker

2

## Özyineleme - örnek

Paskal Üçgeni Dışarıdan girilen sayıyı  
Pascal üçgeninin satır sayısı olarak  
alan ve ilgili üçgene ait sayıları  
ekrana yazdıran (Üçgensel formda)  
programı yazınız.

```

Sayı gir= 4
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
Devam etmek için bir tuşa basın . . .
    
```

```
#include <iostream.h>
#include <iomanip.h>
```

```
long hesapla(int n, int r) {
    if ((n == 0) || (r == 0) || (n == r))
        return 1;
    else
        return (hesapla(n-1,r-1) +
                hesapla(n-1,r));
}
```

```
main() {
    cout<<"Sayı gir= ";
    int num;
    cin>>num;
    for (int i = 0; i<=num; i++) {
        for (int j = num; j > i; j--) cout<<" ";
        for (int k = 0; k <= i; k++)
            cout<<setw(6)<<hesapla(i,k);
        printf("\n");
    }
    system("pause");
}
```

Dr. Şenar Turker

3