

## Veri Yapıları

Kitap: Veri Yapıları ve Algoritmalar, Rifat Gölkesen

C ile Veri Yapıları, Prof.Dr. İbrahim Akman

e-mail: hakankutucu@karabuk.edu.tr

Asistan: Caner Özcan

## Özyinelemeli Fonksiyonlar (Recursive Functions)

Kendi kendini çağıran fonksiyonlardır.

Bir problem: benzer şekilde olan daha küçük parçalara bölerek çözülmesi sağlayan bir tekniktir.

Bir fonksiyon

- 1) iteratif (döngüler) veya
- 2) özyinelemeli olabilir.

### Özyinelemeli bir çözüm için

- 1) Bir temel durum (base case) tanımlanmalı.
- 2) Bir özyinelemeli durum (recursive case) "

### Bir özyinelemeli fonksiyonun genel yapısı

if ( base case )

özyinelemesiz hesapla

else //recursive case

problemi aynı formda daha küçük problemlere böl.

Bunları özyinelemeli çöz.

**Örnek:** Faktöriyel

$$4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$$

$$3! = 3 \cdot 2 \cdot 1 = 6$$

$$2! = 2 \cdot 1 = 2$$

$$1! = 1$$

$$0! = 1$$

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

$$n! = \begin{cases} 1 & \text{eğer } n=0 \\ n \cdot (n-1)! & \text{eğer } n>0 \end{cases}$$

**Soru:** Verilen bir sayının faktöriyelini **özyinelemeli** olarak hesaplayan bir fonksiyon yazınız.

```
int faktoriyel(int n){
    if (n==0)
        return 1;
    else
        return n*faktoriyel(n-1);
}
```

$$\text{faktoriyel}(4) = 24$$

$$\downarrow$$

$$4 \cdot \text{faktoriyel}(3)$$

$$\downarrow$$

$$3 \cdot \text{faktoriyel}(2)$$

$$\downarrow$$

$$2 \cdot \text{faktoriyel}(1)$$

$$\downarrow$$

$$1 \cdot \text{faktoriyel}(0)$$

$$\downarrow$$

$$1$$

**Soru:**

```
int hesapla (int x){
    printf ("%d", x);
    if ( x < 9)
        hesapla ( x+1);
    printf ( "%d", x);
}

main () {
    hesapla (1);
}
```

**Çıktı:** 123....8998.... 1

