

# Temel Programlama II

## Özyinelemeli Fonksiyonlar

Furkan : İSİM  
Kaya : SOYİSİM

191216002 : NUMARA

Merhaba,

Bu raporu 09/05/2020 tarihinde BIL102 kodlu Temel Programlama 2 dersinde verilmiş olan özyinelemeli hesaplama ödevi için hazırladım. İyi okumalar dilerim.

### Yapılan tanımlamalar:

`#define ESAYISI 5` Dizinin kaç elemanlı olacağı bir sabit olacağı için ve çoğu işlemde kullanılacağı için işe bunu sabit sayı olarak tanımlayarak başladım.

Hemen sonrasında kullanıcıdan aldığımız sayıları tutacak ve recursive fonksiyonların üzerinde işlem yapacağı dizimizi sabit sayımızın uzunluğunda tanımladım.

`float elemanlar[ESAYISI];`

### ToplamHesapla():

```
float ToplamHesapla(int sayi, float toplam){  
    if (sayi <= 0) return toplam;  
    toplam += elemanlar[sayi-1];  
    return ToplamHesapla(sayi-1, toplam);  
}
```

Toplama işleminin sağlamasını daha rahat yapabileceğim için bu fonksiyonu yazarak başlamak istedim. Hatta elemanlar dizisini işe dahil etmeden önce recursive olup

olmadığını test etmek için 0 olduğunda çıkan ekrana printf ile sayı değişkenini yazdıran bi fonksiyon yazdım. Daha sonrasında bunun içine toplam değişkenini ekledim ve elemanlar dizisini de yerleştirdim. Böylelikle içine verdiğimiz sayıdan başlayarak 0 a kadar ilerleyen ve

her seferinde kendisinden aşağıdaki elemanların toplamı ile kendisini toplayan bir fonksiyon elde ettim.

## CarpimHesapla():

Bu fonksiyon aynı ToplamHesapla() mantığında çalışıyor tek farkı şu elemanları toplamak yerine çarpıyoruz.

```
float CarpimHesapla(int sayi, float carpim){
    if (sayi <= 0) return carpim;
    carpim *= elemanlar[sayi-1];
    return CarpimHesapla(sayi-1, carpim);
}
```

## SayilariAl():

```
void SayilariAl(int gidilecek, int simdiki){
    if (simdiki >= gidilecek) return;
    printf("%d. sayıyı giriniz: ", simdiki+1);
    float alinanSayi=0;
    if (scanf("%f", &alinanSayi) == 1){
        elemanlar[simdiki] = alinanSayi;
        SayilariAl(gidilecek, simdiki+1);
    }else{
        while (fgetc(stdin) != '\n');
        printf("Hatalı giriş. Lütfen tekrar giriniz.\n");
        SayilariAl(gidilecek, simdiki);
    }
    return;
}
```

Ödevde bu fonksiyonu recursive yapmamız gerektiği belirtilmemiş ama yapmışken neden tüm fonksiyonlarımı recursive yazmayayım ki dedim ve fonksiyonu oluşturmaya başladım.

Sayıları kullanıcıdan nasıl almamız gerekiyor? 1 2 3 4 5 diye n. sayıya kadar. O halde fonksiyon her

çalıştığında nerede olduğunu ve nereye gideceğini bilmeli. Bunu da içine aldığı gidilecek ve simdiki değişkenleri ile çözdüm. Fonksiyonun en başında simdiki gidileceğe ulaşmış mı bunu kontrol ettiriyorum eğer ulaşmış ise return ile fonksiyonumu bitiriyorum. Eğer ulaşmamış ise printf ile sayıyı giriniz yazdırıyorum ve scanf ile ekrandan sayıyı okutuyorum. Scanf eğer sayıyı okuyabiliyorsa bana 1 döndürüyor ve aldığım elemanı elemanlar listemde yerine yerleştirip bir sonraki sayıyı alması üzerine simdiki+1 yaparak fonksiyonu tekrar çalıştırıyorum. Else kısmına gelecek olursam. Mantıken yapılacak olan nedir? Şimdiki sayıyı okumakta sıkıntı çekersek şimdikini tekrar istemeliyiz değil mi? Yani fonksiyonumu o andaki şimdiki değeriyle tekrar çağırمام yeterli olur ama maalesef c de işler farklı yürüyormuş. Bunu yaptığım zaman scanfın içi temizlenmiyordu ve fonksiyon sonsuz döngüye giriyordu.

Buradaki hatayı <https://stackoverflow.com/questions/5062646/scanf-misbehaving> şu kaynağı inceleyerek çözdüm. Bunu ayrı bir başlık altında anlatmak istiyorum. Scanfin okuduğu değeri boş çıkararak fonksiyonu şimdiki değeriyle çağırıp aynı sayıyı tekrar istettim. Böylelikle hatalı durumda da fonksiyon değerleri okuyabilir hale geldi.

## Main():

```
int main(void) {  
    SayilarAl(ESAYISI, 0);  
    printf("Girilen sayıların çarpımı: %.2f\n", CarpimHesapla(ESAYISI, 1));  
    printf("Girilen sayıların toplamı: %.2f\n", ToplamHesapla(ESAYISI, 0));  
    return 0;  
}
```

Olmazsa olmaz main fonksiyonumuz da sırasıyla sayıları alıyor ve sırasıyla diğer

fonksiyonlarımızı çağırıyor. Toplam fonksiyonunu 0 dan başlatarak çağırıp, çarpım hesapla fonksiyonumu 1 den başlatmamın sebebi şu; toplamada 0 etkisiz elemanken, çarpmada 1 etkisiz elemandır.

## Scanf() hatalı davranışı:

C dilinde konsoldan değer alıp değer bastırmak akış mantığı ile gerçekleşiyor yani konsoldan değeri bir sıra halinde alıyoruz aynı şekilde konsola değer yazarken sıra halinde yazdırıyoruz. Scanf fonksiyonu başarılı durumda 1 döndürerek sırada bulunan bilgi parçasını istediğimiz değişkene aktaran bir fonksiyon. Peki ya hatalı durumda ne yapıyor? 0 değerini döndürerek işlemi bitiriyor. Sırada bulunan bilgiyi korumak için de sıraya karışmıyor.

Biz eğer bu hatalı değeri sıradan çıkartmadan bir daha scanf i çağırmaya çalışırsak her zamanki yaptığı işi yapacak, sırada bulunan değeri okumaya çalışacak ve yine hatalı değer döndürecek böyle böyle sonsuza kadar gidecek. Bizim bu sıradan -stdin- hatalı değeri çıkarmamız gerekiyor. Bunun için de while döngüsü ile yeni satırı görene kadar sırayı ilerletebiliriz. `While (fgetc(stdin) != '\n')` Bu komutu kullanarak sıramızda bulunan değerleri teker teker okuyup boşa atıyoruz ve istenilecek yeni değer için sıramızı boşaltmış oluyoruz.