Rekürsif Fonksiyonlar (Özyinelemeli fonksiyonlar)

Kendi kendini çağıran fonksiyonlara rekürsif fonksiyon adı verilir.

Burada dikkat edilmesi gereken nokta, fonksiyonun sonsuz döngüyegirmesini önlemek için uygun bir koşulla sonlandırılması gerektiğidir.

Rekürsif fonksiyonların kullanımı bazı problemlerde programcıya oldukça kolaylık sağlar. Fakat fonksiyonun her kendini çağırması, eski parametre, değişkenlerin saklanması ve geriye dönük sıralı olarak işlem yapılması programın daha fazla hafızaya ihtiyaç duymasına ve daha yavaş çalışmasına yol açacaktır.

Rekürsif Fonksiyon ile Faktöryel Hesabı

```
int faktoryel (int sayi)
{
      if(sayi==0) return 1;
      else return sayi*faktoryel(sayi-1);
}
```

Girilen sayı değerine kadar olan sayıları toplayan program

```
int topla (int sayi)
{
      if(sayi==1) return 1;
      else return sayi + topla(sayi-1);
}
```

Geçmiş Sınav Soruları :

```
int myo (int a, int b, int c)
{
    if(a%b == c)  return a;
    else return  a%b * myo(a-1, b+1, c);
}
main()
{
    printf("%d",myo(10,4,0));
}
```

Çıktı = ?

```
int fonk (int x, int y)
{
    if(x%y == 0)  return  y;
    return  (x+y) * fonk(x+1, y-1);
}
main()
{
    printf("%d",fonk(6,4));
}
```

Çıktı = ?

```
int fonk (int a, int b)
{
    if(sqrt(a) == b)  return  a+b;
    return  b * fonk(a-2, b-1);
}
main()
{
    printf("%d",fonk(13,5));
}
```

Çıktı = ?

```
int gs(int x)
{
    if(x == 1)  return  x;
    return  x +gs(x/3)
}
main()
{
    printf("%d",gs(27));
}
```

Çıktı = ?

C Makroları

C dilinde kullanılan önişlemci komutları şunlardır:

```
#include      #define       #ifdef
#error        #else         #ifndef
#if           #elif         #endif
```

#include: Bu önişlemci verilen dosyanın içeriğini, kullanıldığı yerde kaynak dosyanın içine ekler.

murat.h

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

```
#include <stdio.h>
#include <conio.h>
#include "murat.h"
```

#define: Bu önişlemci komutu, kaynak dosyada bir isim yerine başka bir ismin yerleştirilmesini sağlar.

Programda kullanılan bu sembolik isimler başta ana program olmak üzere bütün alt programlarda da aynı değere sahiptir.

```
#define PI 3.1415926
#define BASLA  {
#define SON  }
#define yaz  printf
#define oku  scanf
```

Örnek kod:

```
#include<stdio.h>
#define PI 3.14

float daire_alan(int yaricap) {
    return PI*yaricap*yaricap;
}

main()
{
    int ycap;
    printf("Dairenin yaricapini girin: ");
        scanf("%d",&ycap);
    printf("\nAlan = %f",daire_alan(ycap));
}
```