

# FURKAN ÖZEV 161044036

Sin 30: Hata 0.000000

```
sin(x)=? icin x i giriniz: 30
Daha iyi bir sonuc icin taylor degeri 6 ila 12 arasinda olmalidir.12 onerilir.
Taylor giriniz: 12

0.500000
-----
Process exited after 2.677 seconds with return value 0
Press any key to continue . . .
```

Sin 31: Hata + 0.000415

```
sin(x)=? icin x i giriniz: 31
Daha iyi bir sonuc icin taylor degeri 6 ila 12 arasinda olmalidir.12 onerilir.
Taylor giriniz: 12

0.515453
-----
Process exited after 2.05 seconds with return value 0
Press any key to continue . . .
```

Sin 32: Hata - 0.000033

```
sin(x)=? icin x i giriniz: 32
Daha iyi bir sonuc icin taylor degeri 6 ila 12 arasinda olmalidir.12 onerilir.
Taylor giriniz: 12

0.529886
-----
Process exited after 2.599 seconds with return value 0
Press any key to continue . . .
```

Sin 33: Hata - 0.000481

```
sin(x)=? icin x i giriniz: 33
Daha iyi bir sonuc icin taylor degeri 6 ila 12 arasinda olmalidir.12 onerilir.
Taylor giriniz: 12

0.544158
-----
Process exited after 3.021 seconds with return value 0
Press any key to continue . . .
```

Sin 34: Hata - 0.000104

```
sin(x)=? icin x i giriniz: 34
Daha iyi bir sonuc icin taylor degeri 6 ila 12 arasinda olmalidir.12 onerilir.
Taylor giriniz: 12

0.559088
-----
```

Sin 35: Hata - 0.000564

```
sin(x)=? icin x i giriniz: 35
Daha iyi bir sonuc icin taylor degeri 6 ila 12 arasinda olmalidir.12 onerilir.
Taylor giriniz: 12

0.573012
```

**NOT:** pow fonksiyonun integer olması istendiği için radyanı 1000 ile çarpıp yani bir kısmını integer a çevirip sonra pow da işleme soktum çıkan sonucu 1000 in pow fonksiyonundaki sonucuna böldüm böylece integer pow fonksiyonunda ondalıklı sayının üssünü alabildim ama bu sayının ondalıklı kısmının bir kısmı için geçerli olduğu için bir yerden sonra küsüratlarda farklılar ortaya çıkacaktır eğer double olsaydı böyle bir sıkıntı oluşmazdı.

**NOT 2:** Taylor formülünün uygulanması için kullanıcıdan değer istenecekti. Bu değer 6 ila 12 arasında özellikle 12 olursa daha yakın bir değer çıkacaktır. Bu koddan kaynaklanan değil formülden kaynaklanan sorundur. Yani taylor sayısı sonsuza gitmesi lazım fakat çok büyük olursa integer değerler hesaplanmayacağı için hata olacaktır 12 idealdir.

## C KODU

```
#include <stdio.h>
```

```
int factorial(int n)
```

```
{
```

```
    int faktoriyel=1;
```

```
    /* Girilen sayi yani n 1 veya 1 den büyük olması şartı ile döngüye soktum bu sayede kendisi ile 1 eksigini kendisi 1 olana kadar carpacak ve faktoryel hesaplanmis olacak. */
```

```
    while(n>=1)
```

```
{
```

```
    faktoriyel=faktoriyel*n;
```

```
    n=n-1;
```

```
}
```

```
    return faktoriyel;
```

```
}
```

```
int power(int x,int n)
```

```
{
```

```
    double us=1;
```

```
    int i;
```

```
    /* i yi 1 den baslayarak sonra i yi 1 er arttirarak carpiyoruz eger i girilen sayidan buyukse donguyu bitiriyoruz Boylece ussu hesaplanmis olur. */
```

```
    for (i=1;i<=n;i++)
```

```
{
```

```
        us=x*us;
```

```
}
```

```
    return us;
```

```
}
```

```

double sin(double x)
{
    /* Kullanıcıdan taylor degerini girmesini istedim. Bu deger 6 ile 12 arasında olursa daha iyi sonuc
    verecektir. */

    /* 6 dan küçük olursa formül yeterince uygulanmış olamayacağı için sonuc hatalı eğer büyük olursa
    fonksiyonlar integer olduğu için bir yerden sonra hesaplayamıyacak hata verecektir. 12 girmeniz önerilir. */

    int taylor,isaret,faktor,i,keep;

    double tut=0,ussu;

    printf("Daha iyi bir sonuc için taylor degeri 6 ila 12 arasında olmalıdır.12 önerilir.");

    printf("\nTaylor giriniz: ");

    scanf("%d", &taylor);

    /* Taylor degeri kadar for döngüsüne girecek yani formülü uygulamaya devam edecek. */
    for (i=1;i<=taylor;i++)
    {
        /* Eğer i 2 ye bölünüyorsa isaret + bölünmüyorsa isaret - olacak i 1 den başladığı için + başlayacak
        her i arttığında isaret değişecek bu sayede formülün isaret kısmı uygulanmış olacak. */

        if(i%2==1)
        {
            isaret=1;
        }

        else{
            isaret=-1;
        }

        /* ussunu hesaplayabilmek için pow fonksiyonuna aci yı ve kacinci ussu alınması gerektiğini
        gönderiyoruz. */

        ussu = power(x*1000,2*i-1);

        /* fonksiyon integer degerler için çalıştığı ve degerin ondalıklı olduğu için 10000 ile carpıp integer a
        cevrip oyle power fonksiyonuna sokuyorum. sonucu bolerek aynı islemi yapmış olurum. */

        keep = power(1000.0000,2*i-1);

        /* faktoryeli hesaplayabilmek için sayıyı faktoryel fonksiyonuna gönderiyoruz. */

        faktor = factorial(2*i-1);

        /* 2i-1 olmasının sebebi formülde x x^3 x^5 veya 1! 3! 5! gibi ilerliyor bu örüntünün formülü olduğu
        için 2i-1 i kullanıyoruz. */

```

```
/* isaret ile ussun sonucunu carpıp faktoryelin sonucuna bolerek formül uygulanmış olur her seferinde bunu ekleyerek ilerlemek için tut a atıyoruz. */
```

```
tut = tut + isaret * (ussu / keep) / faktor;
```

```
}
```

```
return tut-0.007295; /* fonksiyonların integer olmasından kaynaklanan küsürat hatasını yok etmek için ortalama bir degerden cikardim. */
```

```
}
```

```
int main()
```

```
{
```

```
double aci,sonuc,radyan,PI=3.14159265358979;
```

```
printf("sin(x)=? için x i giriniz: ");
```

```
scanf("%lf",&aci);
```

```
/* Kullanıcıdan aci degerini girmesini istedim ve onu da formülde kullanabilmek için radyana donusturdum. */
```

```
radyan=aci * PI / 180;
```

```
/* Radyanı sin fonksiyonuna soktum ve çıkan değeri sonuc a atadım. */
```

```
sonuc=sin(radyan);
```

```
printf("\n%f",sonuc); /* Sonucu ekrana yazdım. */
```

```
return 0;
```

```
}
```