

## Taylor Series

```
double e(int x, int n)
{
    static double p=1, f=1;
    double r;

    if(n==0)
        return 1;
    r=e(x, n-1);
    p=p*x;
    f=f*n;
    return r+p/f;
}

int main()
{
    printf("%lf \n", e(4, 15));
    return 0;
}
```

## Taylor Series Horner's Rule

```
double e(int x, int n)
{
    static double s;
    if(n==0)
        return s;
    s=1+x*s/n;
    return e(x, n-1);
}

int main()
{
    printf("%lf \n", e(2, 10));
    return 0;
}
```

## Taylor Serie Iterative

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

division  
int / int = int  
not double

int  
Type Cast Prevent

$$\left[ \begin{array}{l} x/n * s \\ || \\ s/n * x \\ || \\ x * s/n \end{array} \right]$$

```
double e(int x, int n)
{
    double s=1;
    int i;
    double num=1;
    double den=1;

    for(i=1;i<=n;i++)
    {
        num*=x;
        den*=i;
        s+=num/den;
    }
    return s;
}
int main()
{
    printf("%lf \n",e(1,10));
    return 0;
}
```