```
// e^x = 1 + x/1 + x^2/2! + x^3/3! + x^4/4! +..........

// for making 1 copy of power and factorial we use static variables

// see on the copy for more references




#include <iostream>
using namespace std;




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




// e^x=[1+X [1+X/2] 1+X/3 [1+X/4 * 1] ]   {{ for first 4 element we have this expression}}
double efaster(int x, int n)
{
    static double sum;
    if (n == 0)
    {
        return sum;
    }
    // it is called on the time of going i.e from 4-0; {{ as on the expresion above we have to
find (( 1+x/4*1 )) first }}


    // s=1+x/n*s {{ x,n are integers and give integer division,so to make it double division
,multiple the s with x before dividing with n}}
    sum = 1 + x * sum / n;
    return e(x, n - 1);
}




// here by loops we represent the taylor series i.e metioned on the 1st example
double eloops(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()
{
    // cout << e(1, 10) << endl;
    cout << efaster(1, 10) << endl;
    // cout << eloops(1, 10) << endl;
    return 0;
}
```