

Other helpful hints:

If you look at each of the terms in this series at t_0 , t_1 , t_2 , etc., you may derive each term from it's previous term:

$$t_0 = 1/0! = 1$$

$$t_1 = 1/1! = 1/1 = 1 \quad \text{or} \quad t_1 = t_0/1$$

$$t_2 = 1/2! = 1/(2*1) = 0.5 \quad \text{or} \quad t_2 = t_1/2$$

$$t_3 = 1/3! = 1/(3*2*1) = 0.166667 \quad \text{or} \quad t_3 = t_2/3$$

$$t_4 = 1/4! = 1/(4*3*2*1) = 0.0416667 \quad \text{or} \quad t_4 = t_3/4$$

$$t_n = 1/n! \quad \text{or} \quad t_n = t_{(n-1)}/n$$

It is not necessary (and very time consuming) to generate each term from scratch. Instead generate a newterm based on the previous term.

A possible main program for this assignment (MAXDIG and maybe PRTDIG, along with appropriate includes/prototypes/functions need to be added):

```
void main()
{
    int Term[MAXDIG], Sum[MAXDIG], NewTerm[MAXDIG], N=0;

    ZeroArray(Term);
    ZeroArray(Sum);
    Term[0]=1;
    Sum[0]=1;
    printf("N=%3d -----\n", N);
    printf("TRM="); PrintArray(Term); printf("\n");
    printf("SUM="); PrintArray(Sum); printf("\n");

    for (N=1; N<NUMTRM; N++)
    {
        DivArrayScaler(Term, N, NewTerm);
        CopyArray(Term, NewTerm);
        AddArray(Sum, Term, Sum);

        printf("N=%3d -----\n", N);
        printf("TRM="); PrintArray(Term); printf("\n");
        printf("SUM="); PrintArray(Sum); printf("\n");
    }
}
```

A simple (no arrays) program to calculate e within the limits of double's is below:

```
#define _CRT_SECURE_NO_DEPRECATED
#include <stdio.h>
#define NUMTRM 40

void main()
{
    double Term, Sum;
    int N=0;

    Term=1.0;
    Sum=1.0; // .....1...../.....2
    printf("N=%3d -----\n", N);
    printf("TRM=%20.181f\n", Term);
    printf("SUM=%20.181f\n", Sum);

    for (N=1; N<NUMTRM; N++)
    {
        NewTerm = Term / (double) N;
        Term = NewTerm;
        Sum = Sum + Term;
        // .....1...../.....2
        printf("N=%3d -----\n", N);
        printf("TRM=%20.181f\n", Term);
        printf("SUM=%20.181f\n", Sum);
    }
}
```

NOTE: The program above represents the algorithm. Submitting this code as is will give you a grade of zero. You need to implement this algorithm using a "bignum" type of data.