Lab 3 - TAYLOR SERIES

Write C code that calculates the $y = \sin x$ and $g = \cos x$ functions by using Taylor Series:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots = \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots = \sum_{k=0}^{n} \frac{(-1)^k x^{2k}}{(2k)!}$$

The program,

- Get the **x** value (in radians) from the user ($-\pi \leq x \leq \pi$)**.** (5 pts)
- Get the **n** value (maximum number of iterations) from the user. (5 pts)
- The exact values (**sinXexact** and **cosXexact**) will be calculated and **printed** by the sin(x) and cos(x) functions from math.h. (15 pts)
- Write the nested loop for the factorial calculation of the denominator (**k!**). The program must **print the result of the factorial in each loop** (see the example result image in below). (20 pts)
- Taylor series approximation of y=sin(x) and g=cos(x) functions (**sinXapprox** and **cosXapprox**) will be printed after the maximum iterations. (40 pts)
- The program must **print** the **absolute error value** between the exact and the approximated values as, (15 pts)

$$\mathbf{fabs(sinXexact - sinXapprox)} \text{ and } \mathbf{fabs(cosXexact - cosXapprox)}$$

(**fabs()** function gives the absolute of the argument)

- **Hint-1:** Use pow function for $x^k = pow(x, n)$.
- **Hint-2:** Use "%" for modulo operation. (The modulo (or "modulus" or "mod") is the remainder after dividing one number by another.)
- **Warning**: In math.h library, cos() and sin() functions take the argument of radians.
- **Warning:** Do not forget the type casting (or type-conversion)

Example Output:

```
Enter the value of n (maximum iteration number)
: 5
Enter the value of x in radians
: 0.5
Exact value of sin(x) = 0.479426
Exact value of cos(x)= 0.877583
0!=1.000000
1!=1.000000
2!=2.000000
3!=6.000000
4!=24.000000
5!=120.000000
n = 5:
sin(x) = 0.479427
cos(x) = 0.877604
sinXexact - sinX = 0.000002
cosXexact - cosX = 0.000022
```