Funciones

1) Escribir una función que dados dos números devuelva el mínimo de ellos.
2) Escribir una función que dado un número devuelva su valor absoluto, sin usar la librería matemática.
3) Escribir cuatro funciones tales que dados dos números complejos (en forma cartesiana), calcule la +,-,* y /. Utilice en todo momento internamente la representación cartesiana, no pase los números a polares. Utilice paso por referencia para los argumentos de salida. Posteriormente escriba una única función con un argumento más (un char) que defina la operación a realizar, devolviendo esta función el resultado que corresponda a la operación especificada.
4) Escribir una función que sume de los n primeros números naturales, n se pasa a la función como argumento.
5) Escribir una función que calcule el producto de los n primeros números naturales, n se pasa a la función como argumento.
6) La cte. de Euler se define como $C = \underset{n\to\infty}{Lim}(1+1/2+1/3+...+1/n-\ln(n))$.

   a) Escribir una función que la calcule de forma aproximada para un valor de n que considere suficiente, n se pasa a la función como argumento.
   b) Escribir una función que la calcule de forma aproximada, según una tolerancia que considere suficiente. La tolerancia se pasa a la función como argumento.

7) Aproximar "e" por la sucesión 1+1/1!+1/2!+1/3!+1/4!... escribir tres funciones, en cada caso pasar los argumentos adecuados a la función correspondiente:
   a) Número fijo de iteraciones.
   b) Precisión máxima.
   c) Lo que suceda antes.
8) Aproximar e^x por la sucesión 1+x/1!+x^2/2!+x^3/3!+x^4/4!... escribir tres funciones, en cada caso pasar los argumentos adecuados a la función correspondiente:
   a) Número fijo de iteraciones.
   b) Precisión máxima.
   c) Lo que suceda antes.
9) La raíz n-ésima de un número puede aproximarse mediante la sucesión definida por la formula recurrente: $x_i = \frac{1}{n}\left((n-1)x_{i-1} + \frac{A}{x_{i-1}^{\,n-1}}\right)$ en donde A es el número del cual quiere hallarse su raíz n-ésima. Particularizando para el caso n=2 (raíces cuadradas), escribir tres funciones que aproximen la raíz cuadrada de un número mediante esta sucesión, en cada caso pasar los argumentos adecuados a la función correspondiente:
   a) Número fijo de iteraciones.
   b) Precisión máxima.
   c) Lo que suceda antes
10) Abundando en este tema, escribir una función que calcule la raíz n-ésima de un número. La funcion tendrá como argumentos el radicando, el orden de la raíz, la tolerancia y el máximo número de iteraciones a realizar. Modifique la función anterior de modo que desde dentro del programa principal se tenga el conocimiento de por cual de las dos razones (iteraciones máximas o tolerancia) ha terminado la función.
11) Escribir una función que devuelva la suma de las cifras de un número expresado en base 10.
12) Escribir una función que devuelva un número expresado en base 10 del revés. Usando esta escriba una función que devuelva "1" o "0" en función de si el número es o no es capicúa. Esta segunda función debe llamar a la primera.
13) Escribir una función que convierta un número de binario a decimal (el número binario se introduce como entero).
14) Escribir una función que calcule el término "n" de la sucesión de Fibonacci, "n" se pasa como argumento a la función. Una vez realizada esta función, escríbala en forma recursiva, explique porque esta implementación realiza más operaciones que la primera. La sucesión de Fibonacci se define como:

$$\begin{cases} f_0 = 1 \\ f_1 = 1 \\ f_i = f_{i-1} + f_{i-2} \text{ si } i \geq 2 \end{cases}$$

15) Escribir una función que calcule el factorial de un número. Utilizando esta función escriba otra que calcule el número combinatorio "n sobre m". Una vez resuelto este problema de esta forma, escriba una

función diseñada específicamente para que calcule dicho número combinatorio, de modo que se realice el menor número posible de operaciones.