

Los archivos generados deben respetar el siguiente formato de nombre `guiaDeClase04_ejercicioNumero.c`. Por ejemplo el archivo del ejercicio 1 debe llevar el nombre `guiaDeClase04_01.c`. Si el ejercicio tuviera ítems a y b por ejemplo el nombre deberá ser `guiaDeClase04_01_A.c` para el punto A. Todos los archivos deberán ser subidos al repositorio dentro de una carpeta con el nombre `guiaDeClase04`. Cada archivo deberá contener la función con el nombre indicado en el ejercicio y un `main` para comprobar su funcionamiento. Deberá comentar cada archivo como se indica en el ejemplo adjunto con esta guía.

1. Implemente una función que imprima la leyenda "Hola Mundo" por `stdout`. El prototipo de la función es el siguiente `void imprimeHolaMundo (void)`
2. Implemente una función que devuelve
  - o El número 5. El prototipo de la función es `int devuelve5 (void);`
  - o La letra A. El prototipo de la función es `char devuelveA (void);`
3. Implemente una función que devuelve el número pasado como parámetro sumándole 1. El prototipo de la función es: `int suma1 (int a);`
4. Compile, linkee y ejecute el siguiente programa. Explique por qué la variable `a` y `b` del `main` no se ven afectadas después de llamar a la función `incremento`.

```
#include <stdio.h>
int suma1(int a);
int main (void)
{
    int a, b, r;
    //-- Inicializo --
    a = 0; b = 0; r = 0;
    printf ("En el main: a = %d\tb = %d\tr = %d\r\n", a, b, r);

    r = suma1(a);

    //-- Imprimo los valores --
    printf ("En el main: a = %d\tb = %d\tr = %d\r\n", a, b, r);

    return (0);
}
int suma1(int a){
    int b;
    b = 0;
    printf ("En la funcion: a = %d\tb = %d\r\n", a, b);
    a++; b++;
    printf ("En la funcion: a = %d\tb = %d\r\n", a, b);
    return (a);
}
```

5. Implemente una función devuelva la suma de dos números enteros pasados como parámetros con un prototipo como el siguiente `int sumaDosNumeros (int a, int b)`
6. Implemente dos funciones una que calcule el área de un círculo y otra su perímetro. Utilice la constante de `pi` de `math.h`. Los prototipos son `float areaCirculo (float radio);`  
`float perimetroCirculo (float radio);`
7. Implemente una función que calcule el valor absoluto de un número con tipo de dato `float`. No utilice las funciones de `math.h` Prototipo: `float valorAbsoluto (float valor);`
8. Implemente una función que devuelva el valor máximo entre dos números pasados como parámetros. Prototipo: `int valorMaximo (int a, int b);`
9. Implemente una función a la cual le pase un carácter como parámetro y me devuelve
  - 0: si el carácter es una letra mayúscula.
  - 1: si el carácter es una letra minúscula.
  - 2: si el carácter es un número.
  - 3: en caso que no sea ninguno de los anteriores.Utilice defines para las constantes. El prototipo es `int filtroASCII (char caracter);`
10. Implemente una función que realice las cuatro operaciones básicas entre dos números de tipo `float` y retorne el resultado. El prototipo de la función es el siguiente  
`float calculo (float opA, float opB, char op)`  
Donde:
  - `opA` y `OpB` son los números con los cuales se debe realizar la operación
  - `op`: La operación a realizar
    - '+': Realiza la suma.
    - '-': Realiza la resta.
    - '\*': Realiza el producto
    - '/': Realiza la división
  - La función retorna el resultado o `NaN` en caso de que se ingrese una operacion no valida.
11. Implemente una función que le pase como parámetro dos números que representan los catetos de un triángulo rectángulo y me devuelva la hipotenusa. El prototipo es `float calcHipo (float catetoA, float catetoB)`
12. Implemente una función que calcule el factorial de un número pasado como parámetro. Si el factorial no puede ser calculado la función debe devolver cero. `unsigned int factorial (unsigned int n);`
13. Implemente un programa que calcule  $x^y$  sin utilizar la función `pow`. El prototipo de la función es `float myPow (int x, int y);`