

4. Trabajo Práctico 4 - Funciones

4.1. Ejercicio 1

Realizar una función que calcule el factorial de un número entero pasado como argumento. Se deben verificar todas las condiciones correspondientes para que el resultado sea correcto, en caso no poder devolver el resultado correcto, retornar 0 (cero). Utilice el siguiente prototipo de función:

```
unsigned long factorial(unsigned int numero);
```

4.2. Ejercicio 2

Realice una función que reciba como argumento un número entero y determine si el mismo es **primo**. La función debe retornar:

- 1 (uno) si es primo.
- 0 (cero) si no es primo.

Utilice el siguiente prototipo de función:

```
int verificar_primo(unsigned int numero);
```

4.3. Ejercicio 3

Escriba una función que genere un número entero pseudoaleatorio en el rango definido por dos números enteros pasados como argumentos. Recordar verificar que el mínimo sea menor al máximo, caso contrario, retornar el valor 0 (cero). **La función debe poder generar números aleatorios tanto positivos como negativos**. Utilice el siguiente prototipo de función:

```
int generar_numero_aleatorio(int min, int max);
```



Ayuda

! Revise: man rand, srand y time

Utilizar los datos entregados por la función `time()` como semilla para las funciones de pseudoaleatoriedad. Haga un comentario en el código explicando por qué se utiliza una semilla y qué pasa si ésta es constante.

4.4. Ejercicio 4

Implemente una función que calcule x^y **sin utilizar la función estándar `pow()`**. La función debe admitir exponentes enteros positivos, negativos, o de valor 0 (cero). Utilice el siguiente prototipo de función:

```
float my_pow(float x, int y);
```