

## Práctica de listas, rangos y ciclos

### rangos

1. Escribe un tipo `range()` que emita: `[100, 101, 102, 103]`
2. Escribe un tipo `range()` que emita: `[-50, -2050, -4050, -6050]`
3. Escribe un programa que pida un número entero y emita una lista de números consecutivos del 0 al valor dado.
4. Escribe un programa que pida dos números enteros (el segundo mayor que el primero) y emita listas de números consecutivos al derecho y al revés.
5. Escribe un programa que pida dos números enteros y emita la lista de números consecutivos que hay entre ellos, de menor a mayor.
6. Escribe un programa que pida dos números enteros y emita la lista de números pares que hay entre ellos (incluidos ellos mismos si son pares)
7. Escribe tres programas que emitan las siguientes secuencias de números:
  - En el primer programa, el tipo `range()` que se utilice en cada bucle debe tener un único argumento.
  - En el segundo programa, el tipo `range()` que se utilice en cada bucle debe tener dos argumentos.
  - En el tercer programa, el tipo `range()` que se utilice en cada bucle debe tener tres argumentos.

### Listas

1. Escribe un programa que permita crear una lista de palabras. Para ello, el programa tiene que pedir un número y luego solicitar esa cantidad de palabras para crear la lista. Por último, el programa tiene que emitir la lista.
2. Escribe un programa que permita crear una lista de palabras y que, a continuación, pida una palabra y diga cuántas veces aparece esa palabra en la lista.
3. Escribe un programa que permita crear una lista de palabras y que, a continuación, pida dos palabras y sustituya la primera (que debe estar en la lista) por la segunda. Emitir la lista.
4. Escribe un programa que permita crear una lista de palabras y que, a continuación, pida una palabra y elimine esa palabra de la lista.
5. Escribe un programa que permita crear dos listas de palabras y que, a continuación, elimine de la primera lista los nombres de la segunda lista.
6. Escribe un programa que permita crear una lista de palabras y que, a continuación, cree una segunda lista igual a la primera, pero al revés (crear una lista distinta).

### Ciclo for

1. Escribe un programa que le permita realizar la escritura de los primeros 100 números naturales.
2. Escribe un programa que le permita realizar la suma de los primeros N números impares. El N debe ingresarse por teclado.
3. Escribe un programa que calcule el factorial de un número cualquiera que se ingresa por teclado.
4. Muestre los N primeros números de la secuencia de Fibonacci, siendo n un dato entero.

## Ciclo while

1. Escribe un programa que pida la cantidad de números positivos que se tienen que ingresar y a continuación pida números hasta que se hayan ingresado la cantidad de números indicada.
2. Escribe un programa que pida dos números enteros. El programa pedirá de nuevo el segundo número mientras no sea mayor que el primero. El programa termina y emitirá los números.
3. Escribe un programa que pida números decimales mientras el usuario escriba número mayores que el primero.
4. Escribe un programa que pida números enteros mientras el usuario ingresa números cada vez más grandes, el programa emite en cada iteración el número anterior ingresado y finaliza ingresando un número menor.
5. Escribe un programa que pida números mientras no se escriba un número negativo. El programa terminará emitiendo la suma de los números ingresados.
6. Escribe un programa que pida un valor límite positivo y a continuación pida números hasta que la suma de los números introducidos supere el límite inicial.
7. Escribe un programa que pida primero dos números enteros (mínimo y máximo) y que después pida números enteros situados entre ellos. El programa terminará cuando se escriba un número que no esté comprendido entre los dos valores iniciales y emitirá la cantidad de números ingresados.
8. Escribe un programa que pida números pares mientras el usuario indique que quiere seguir introduciendo números. Para indicar que quiere seguir escribiendo números, el usuario deberá contestar 'S' o 's' a la pregunta.