Ejercicio 1

1. Para generar un vector ascendente, primero genera el vector del tamaño máximo que se quiera para el mismo. Luego hace un for desde i = 0 hasta el tamaño máximo del vector menos 1 y va insertando el valor de i en el vector.
2. Para generar un vector descendente, primero genera el vector del tamaño máximo y luego hace un for en el desde i = 1 hasta el tamaño máximo del vector menos 1, y luego va insertando tamaño\_maximo – i en el vector. De esta manera, si tenemos un tamaño máximo de 10, insertará 9, 8, 7 … 0.
3. El algoritmo de generar datos aleatorios genera 2 vectores, el que se devolverá con los datos aleatorios y el que vector llamado datosUtilizados que utilizará para saber si ya se utilizó un valor. Repetirá lo mismo tantas veces como el largo del vector a devolver, generará un número aleatorio mientras ese número aleatorio ya haya sido utilizado (apoyándose en el vector datosUtilizados) modificará el mismo agregándole 1 y haciéndole la operación módulo con el largo del vector para que no se exceda. En caso de que no sea utilizado, simplemente utiliza este número aleatorio como la posición en donde pondrá el valor del índice actual del for.
4. Esto es completamente parametrizable, básicamente utiliza una variable de clase llamada tamanio\_max la cual contiene el tamaño máximo que tendrán los vectores. El problema de parametrizar esta variable sería que cada método podría recibir un valor de tamaño máximo distinto, y justamente no es lo que nosotros queremos. Nosotros queremos que los 3 tipos de vectores tengan el mismo tamaño de manera de poder comparar exactamente si son iguales.
5. Se podría hacer un algoritmo que recorra todo el vector utilizando una variable i que empiece en 0 y termine en el largo del vector menos 1. Utilizando ese i, preguntar si el valor que se encuentra en la posición i-esima es menor que el que se encuentra en la posición i-esima más 1. De esta manera, el algoritmo sería de orden N, siendo N el largo del array.