

# Invierte una lista

En este ejercicio se trata de practicar las listas enlazadas simples. Se leerá una serie de valores de la entrada que se guardan en una lista enlazada simple en orden inverso al de entrada. Es decir si la entrada es 1 2 3 4, en la lista enlazada el primer nodo debe tener el valor 4, el segundo el tres etc. El ejercicio pide que se invierta el orden de los nodos de la lista. Por ejemplo, dada la lista anterior: 4 3 2 1, su inversa será 1 2 3 4.

## *Requisitos de implementación.*

En la implementación del problema, se utilizará la clase *SinglyLinkedList* del ejercicio anterior. La clase se completará con un método público, que invierta la lista.

Es importante que la implementación del método que invierte la lista no reserve memoria dinámica, esto es, no se debe realizar ningún new. Esto supone que no se puede llamar a la función `push_front`, ya que esta reserva memoria.

## Entrada

La entrada consta de una serie de casos de prueba. Cada caso se muestra en dos líneas, la primera contiene el número de elementos de la lista, en la segunda se muestran los elementos.

Los elementos de la lista son números enteros. En este caso el cero puede ser un elemento de la lista y puede haber listas con cero elementos. Si la lista tiene cero elementos se mostrará un cero en la primera línea y la segunda aparecerá vacía.

## Salida

Para cada caso de prueba se escriben dos líneas. En la primera se muestra la lista en el orden que se ha creado, esto es en orden inverso a la entrada de datos, ya que los datos se añaden por el principio. En la segunda línea se muestra la lista invertida. Si la lista de entrada es vacía se dejan dos líneas en blanco.

## Entrada de ejemplo

```
4
5 3 1 8
0

1
7
```

## Salida de ejemplo

```
8 1 3 5
5 3 1 8

7
7
```

**Autor:** Isabel Pita.