

## Ejercicio 2 (2 puntos)

Queremos implementar una función que reciba una lista y la modifique para que tenga solamente la primera aparición de cada elemento diferente.

```
template<typename T>
void eliminar_repetidos(std::list<T> & l);
```

Por ejemplo, si la lista es  $l = [1, 5, 1, 1, 2, 5, 2]$  entonces `eliminar_repetidos(l)` haría que la lista se transformara en  $l = [1, 5, 2]$ . Es decir, solamente se conserva la primera aparición de cada valor, y estos mantienen su orden de aparición.

Implementa esta función, que debe manipular la lista de forma abstracta, es decir, sin acceso a su representación interna.

El coste de la operación debe ser lineal con respecto al número de elementos en la lista. Justifica el coste de tu implementación.

### Entrada

Cada caso de prueba está formado por tres líneas. La primera contendrá el carácter N si los elementos de la lista son números, o el carácter P si los elementos son palabras. La segunda línea contendrá un entero  $L > 0$  que representa la longitud de la lista. La tercera línea contendrá los  $L$  elementos de la lista. Si son números estarán en el rango  $[0..10^9]$ . Si son palabras, estarán formadas por no más de 30 caracteres de la 'a' a la 'z'.

### Salida

Para cada caso de prueba se escribirá una línea con los elementos de la lista después de eliminar los elementos repetidos.

### Entrada de ejemplo

```
N
7
1 5 1 1 2 5 2
P
6
perro gato perro gato perro gata
P
1
uno
```

### Salida de ejemplo

```
1 5 2
perro gato gata
uno
```