

## Destripar una cadena



Supongamos una cadena formada por números y letras mayúsculas, por ejemplo: "023P3T4RD0N98". *Destripar* una cadena consiste en eliminar todos los caracteres numéricos situados en los extremos, hasta llegar al primer carácter alfabético. Por ejemplo, el resultado de destripar la cadena de ejemplo anterior sería "P3T4RD0N".

Una cadena de texto puede verse como una lista de caracteres. En este ejercicio trabajaremos con la clase `ListLinkedListSingle` vista en clase. En este caso, el tipo de los elementos de la lista será `char`. En este ejercicio se pide:

1. Añadir el siguiente constructor a `ListLinkedListSingle`:

```
ListLinkedListSingle::ListLinkedListSingle(const string &texto);
```

Este constructor recibe una cadena de texto y construye una lista cuyos elementos son los caracteres de dicha cadena. Por ejemplo, la siguiente sentencia,

```
ListLinkedListSingle ls("45HOLA2")
```

construye la lista `ls = ['4', '5', 'H', 'O', 'L', 'A', '2']`.

2. Añadir a la clase `ListLinkedListSingle` el siguiente método:

```
void ListLinkedListSingle::destripar(ListLinkedListSingle &dest);
```

Este método elimina los caracteres numéricos de la lista `this` situados en los extremos, y los añade a la lista `dest`, que se supone vacía en el momento de llamar a `destripar`. Los caracteres eliminados deben colocarse en `dest` en el mismo orden en el que aparecían inicialmente en `this`. Por ejemplo, tras ejecutar lo siguiente:

```
ListLinkedListSingle ls("45HOLA2");  
ListLinkedListSingle ms;  
ls.destripar(ms);
```

La lista `ls` contiene los elementos `['H', 'O', 'L', 'A']` y la lista `ms` contiene los elementos `['4', '5', '2']`.

**Importante:** En la implementación del método `destripar` no pueden crearse, directa o indirectamente, nuevos nodos mediante `new` ni eliminarse mediante `delete`. Tampoco se permite copiar valores de un nodo a otro.

3. Indica y justifica el coste del constructor y del método implementados anteriormente.
4. Escribe un programa que lea varias cadenas de entrada y llame al método `destripar` sobre sus correspondientes listas de caracteres, pasando una lista vacía como parámetro `dest`. Para ello utiliza el formato de entrada y salida descrito a continuación.

## Entrada

La entrada comienza por un número que indica cuántos casos de prueba vienen a continuación. Cada caso de prueba ocupa una línea formada exclusivamente por caracteres numéricos y letras mayúsculas. Cada línea puede contener un número de caracteres comprendido entre 1 y 100 000.

## Salida

Para cada caso de prueba deben imprimirse dos líneas: una con la lista resultante tras haber llamado al método `destripar` y otra con la lista pasada como parámetro a este método. Utiliza la notación vista en clase. Puedes llamar al método `ListLinkedListSingle::display()` para ello.

### Entrada de ejemplo

```
5
023P3T4RDON98
45HOLA2
FGD21
235A8D76
A47B31
```

### Salida de ejemplo

```
[P, 3, T, 4, R, D, O, N]
[0, 2, 3, 9, 8]
[H, 0, L, A]
[4, 5, 2]
[F, G, D]
[2, 1]
[A, 8, D]
[2, 3, 5, 7, 6]
[A, 4, 7, B]
[3, 1]
```

## Créditos

**Autor:** Manuel Montenegro