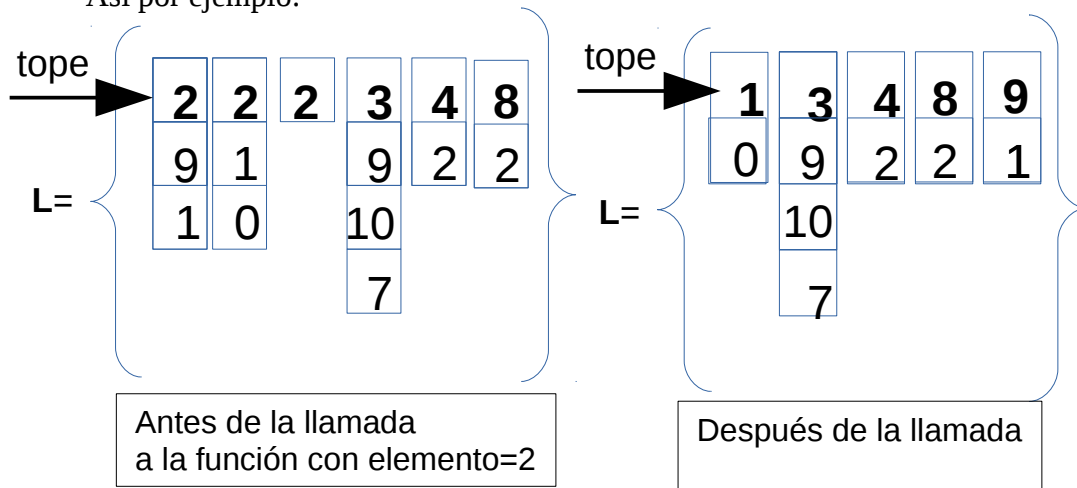


**Estructuras de Datos**  
**Curso 2014-2015. Convocatoria de Febrero**  
**Grado en Ingeniería Informática.**  
**Doble Grado en Ingeniería Informática y Matemáticas**

1. (1 punto) Razonar (en términos de eficiencia de las operaciones) qué TDA elegirías entre los que se indican para:
  - (a) Implementar una **Cola con prioridad** (con funciones básicas de insertar, borrar, y frente) de entre: un **ABB** (árbol binario de búsqueda), un **APO** (árbol parcialmente ordenado implementado mediante un **heap**) y un **Vector ordenado**.
  - (b) Implementar un **Diccionario** (con funciones básicas de insertar, borrar y buscar) de entre: una **lista ordenada**, un **vector ordenado** y un **AVL**.
  - (c) Implementar un **Conjunto no ordenado** (con funciones básicas de insertar, borrar y buscar) de entre: un **AVL**, un **ABB**, y una **Tabla hash abierta**.
2. (1.5 puntos) Dada una lista que contiene pilas con enteros y que se encuentra ordenada de menor a mayor por el tope de cada pila, implementar una función **borrar**, que elimina el tope de cada pila en la lista, con valores iguales al dado como parámetro. La cabecera de la función sería:

***void Borrar(list<stack<int> > &L, int elemento)***

Así por ejemplo:



**Nota:** la lista tiene que quedar ordenada por el tope de las pilas tras la ejecución de la función.

3. (1.5 puntos) Implementar una función que muestre las etiquetas de los nodos de un árbol binario de enteros que estén entre dos **niveles** dados **n1** y **n2** ( $0 \leq n1 < n2$ ) ambos inclusive

4. (1 punto) Se dice que un árbol binario de enteros es inferior a otro si (teniendo la misma estructura de ramificación), los elementos del primero, en los nodos coincidentes en posición, son menores que los del segundo. Diseñar una función booleana que dados dos árboles binarios, devuelva true si el primero es inferior al segundo

***bool es\_inferior(bintree<int> & ab1, bintree<int> & ab2)***

- 5.- (1 punto) Implementar un iterador que itere sobre las claves que sean números primos en una clase Diccionario definida como:

```
class Diccionario{
    private:
        map<int, list<string> > datos;
        .....
        .....
}
```

Han de implementarse (aparte de las de la clase iterador) las funciones begin() y end(). Se supone implementada una función **bool primo (int x)** que devuelve true si el entero x es primo.

**Tiempo: 3 horas**