



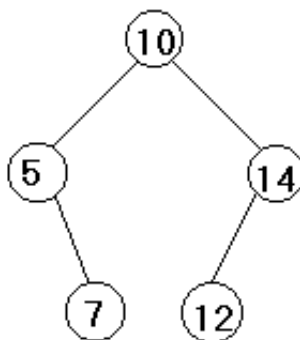
Nombre y Apellido:

Mail:

## Prueba

Para resolver algunos ejercicios de este examen deberá utilizar las funciones de alto orden que vienen definidas ya en ml: **map**, **foldl**, **foldr** y **filter**. No debe redefinirlas. Para resolver algunos de los ejercicios deberá utilizar, además, una combinación de ellas.

1. Dado el siguiente árbol:



representéelo usando:

- a. el siguiente tipo de dato:

```
- datatype 'a arbol1 =  
  Vacio |  
  Nodo of 'a arbol1 * 'a * 'a arbol1;
```

- b. el siguiente tipo de dato:

```
- datatype 'b arbol2 =  
  Nodo of 'b * 'b arbol2 list;
```

2. Teniendo en cuenta las definiciones de árboles del ejercicio anterior:

- Escriba una función que tome como argumento un árbol de tipo **string arbol1** y devuelva un **int arbol1** donde los valores de los nodos del árbol resultado coincida con la longitud de las cadenas del árbol de entrada (o sea, si la raíz del árbol de entrada contiene la palabra "casa", entonces el árbol resultado tendrá un 4 en su raíz).
- Escriba una función que tome un árbol de tipo **real arbol2** y un elemento **x** de tipo **real** y devuelva la cantidad de veces que **x** aparece en el árbol.
- Escriba una función que tome un árbol de tipo **real arbol2** y devuelva el producto de todos sus nodos.

3. Resuelva los siguientes items usando las funciones **map**, **foldl**, **foldr** y **filter** o una combinación de ellas.

- Defina la función **listabs** que tome una lista de reales y devuelva la lista de sus valores absolutos.
- Defina la función **sumneg** que tome una lista reales y devuelva la suma de los elementos negativos.
- Defina la función **contarpalabras** que tome una lista de strings y devuelva la cantidad de strings cuya longitud sea menor estricta que 5.
- Defina la función **filtrarlargas** que tome una lista de strings y devuelva otra lista que contenga sólo los strings de la lista original cuya longitud sea menor estricta a 5.

4. Dado el siguiente tipo de dato:

```
- datatype 'etiqueta arbolbin =  
  Vacio |  
  Nodo of 'etiqueta arbolbin * 'etiqueta * 'etiqueta arbolbin;
```

Suponga que existe un árbol binario de búsqueda de tipo **(int \* int) arbolbin**. Cada nodo del árbol es una tupla que contiene dos números enteros. El árbol está ordenado según la primera componente de cada par. Implemente las siguientes funciones:

- a. Defina una función `listar_en_orden_creciente` que tome un árbol binario de búsqueda `(int * int) arbolbin` y devuelva un `(int * int) list`, respetando el orden creciente de las primeras componentes de los pares. Para ello, deberá definir una función que implemente uno de los recorridos (`inorden`, `preorden` o `postorden`) y usar esa función.
- b. Defina una función `seg_comp_dos` que tome un árbol binario de búsqueda `(int * int) arbolbin` y devuelva un `(int, int) list` que contenga únicamente los pares ordenados del árbol original cuya segunda componente sea 2. Debe usar la función `listar_en_orden_creciente` y `filter` para su definición.