



Nombre y Apellido:

Mail:

Prueba

Resuelva el siguiente examen en la computadora. Guarde todas las soluciones en un único archivo con su nombre y apellido: `NombreApellido.ml`

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 tipo de dato:

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

- Defina la función `preOrden` que realice el recorrido *preorden* de un `arbolbin`
- Defina la función `inOrden` que realice el recorrido *inorden* de un `arbolbin`
- Defina la función `postOrden` que realice el recorrido *postorden* de un `arbolbin`

2. Suponga que representamos puntos en el plano con pares ordenados de reales. O sea, definimos el tipo de dato `par`:

```
- type par = real * real;  
type par = real * real
```

Un ejemplo de un punto en el plano sería el siguiente:

```
- val p=(2.0,3.0):par;  
val p = (2.0,3.0) : par
```

Sabemos que si tenemos un punto (x,y) en el plano, su distancia al origen de coordenadas se calcula así:

$$dist_origen_coordenadas(x,y) = \sqrt{x^2 + y^2}$$

- Defina la función `listdist` que tome una lista de puntos del plano y devuelva la lista de sus distancias al origen de coordenadas.
- Defina la función `sumdist` que tome una lista de puntos del plano y devuelva la suma de sus distancias al origen de coordenadas.
- Defina la función `puntosalejados` que tome una lista de puntos del plano y devuelva otra lista que contenga sólo aquellos puntos que se encuentran a una distancia estrictamente mayor a 5.0 del origen de coordenadas.

3. Considere nuevamente la definición de árbol binario del ejercicio 1.

Suponga que existe una lista de estudiantes con sus notas de un parcial implementada como un árbol binario de búsqueda de tipo `(string * int) arbolbin`. Cada nodo del árbol es una tupla que contiene el nombre del alumno y su nota. El árbol binario de búsqueda está ordenado de acuerdo al orden alfabético del campo nombre (suponga todos los nombres escritos en minúscula y sin acentos). Implemente las siguientes funciones:

- Defina una función `listar_en_orden_creciente` que tome la lista de estudiantes implementada como un árbol binario de búsqueda `(string * int) arbolbin` y devuelva un `(string * int) list` de todos los alumnos con sus notas. Use alguna de las funciones de recorrido definidas en el ejercicio 1.
- Defina una función `buscar_sobresalientes` que tome la lista de estudiantes implementada como un árbol binario de búsqueda `(string * int) arbolbin` y devuelva un `(string, int) list` de todos los alumnos que hayan obtenido un 10 como calificación. Debe usar la función `listar_en_orden_creciente` y `filter` para su definición.