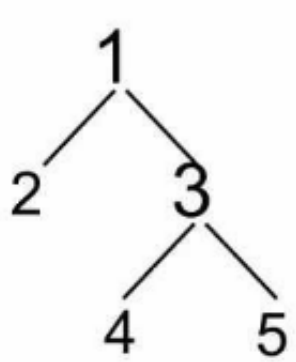
Instituto Politécnico Superior "Gral. San Martín"
Departamento de Informática
5^{to} Año - Programación II

Nombre y Apellido: Mail:

Examen Recuperatorio

Para resolver algunos ejercicos de este examen deberá utilizar las funciones de alto orden currificadas 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:



   represéntelo usando:

   a. el siguiente tipo de dato:

```
- datatype 'a arbol1 =
            Vacio |
            Nodo1 of 'a arbol1 * 'a * 'a arbol1;
```

   b. el siguiente tipo de dato:

```
- datatype 'b arbol2 =
            Nodo2 of 'b * 'b arbol2 list;
```

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

   a. Escriba una función `sumaParesArbol` que tome como argumento un árbol de tipo `int * int arbol1` y devuelva un `int arbol1` donde los valores de los nodos del árbol resultado coincida con la suma de los elementos de cada par ordenado del árbol de entrada (o sea, si la raíz del árbol de entrada contiene el par `(3,4)`, entonces el árbol resultado tendrá un `7` en su raíz).

   b. Escriba una función `aparece` que tome un árbol de tipo `string arbol2` y un elemento `x` de tipo `char` y devuelva `true` si la letra aparece en alguna de las palabras del árbol, `false` en caso contrario.

   c. Escriba una función `cuadrados` que tome un árbol de tipo `int arbol2` y devuelva otro `int arbol2` donde cada elemento del árbol resultado sea el cuadrado del elemento correspondiente en el árbol original (o sea, si la raíz del árbol de entrada contiene un `4` entonces el árbol resultado tendrá un `16` en su raíz).

3. Resuelva los siguientes items usando las versiones currificadas de `map`, `foldl`, `foldr` y `filter` que vienen ya definidas en ML o una combinación de ellas.

   a. Defina la función `listrunc` que tome una lista de reales y devuelva la lista de los valores truncados.

   b. Defina la función `sumpos` que tome una lista reales y devuelva la suma de los elementos positivos.

   c. Defina la función `filtrar2daComponente` que tome una lista de `int * int` y devuelva otra lista que contenga sólo los pares ordenados de la lista original cuya segunda componente sea mayor a 5.

   d. Defina la función `sumarfiltrados` que tome una lista de `int * int`, le aplique la función `filtrar2daComponente` a dicha lista y devuelva el resultado de sumar todas las primeras componentes de la lista resultado.

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 `(string * int) arbolbin`. Cada nodo del árbol es una tupla que contiene una cadena de caracteres y un entero. El árbol está ordenado según la componente entera de cada par. Implemente las siguientes funciones:

a. Defina una función `listar_en_orden_creciente` que tome un árbol binario de búsqueda `(string * int) arbolbin` y devuelva un `(string * int) list`, respetando el orden creciente de las segundas 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 `filtrar_palabras_cortas` que tome un árbol binario de búsqueda `(string * int) arbolbin` y devuelva un `(string, int) list` que contenga únicamente los pares ordenados del árbol original cuya cadena de caracteres tenga una longitud mayor o igual a 5. Debe usar la función `listar_en_orden_creciente` y `filter` para su definición.