



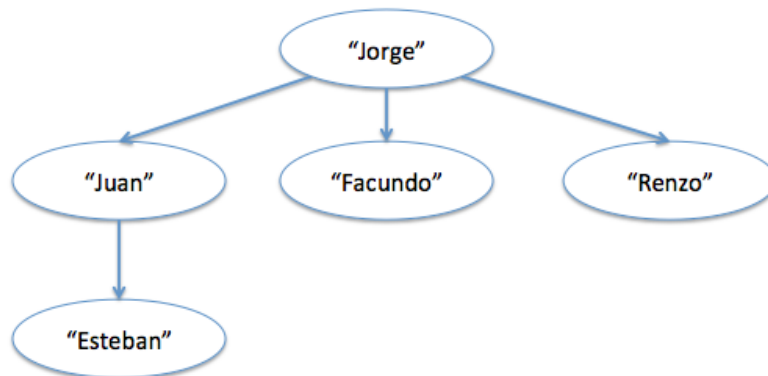
Nombre y Apellido:

Mail:

Examen Final

Cuando se le indique 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. (0.5 puntos) Dado el siguiente árbol:

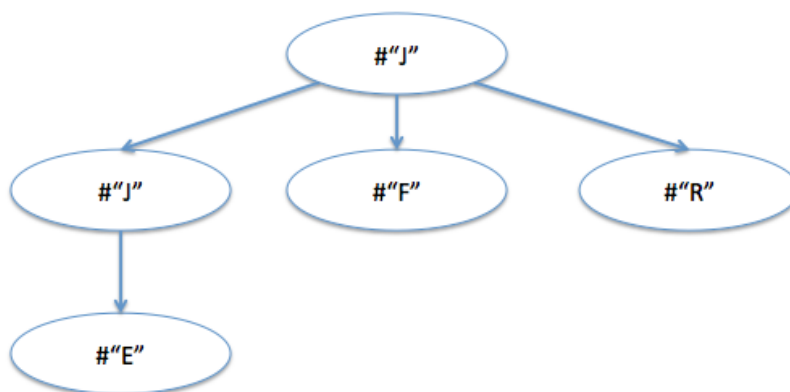


representelo usando el siguiente tipo de dato:

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

2. Teniendo en cuenta la definición de árbol del ejercicio anterior:

- (1 punto) Escriba una función `cantAparece` que tome un árbol de tipo `string arbol2` y un elemento `s` de tipo `string` y devuelva la cantidad de veces que `s` aparece en el árbol.
- (1.5 puntos) Escriba una función `firstLetter` que tome un árbol de tipo `string arbol2` y devuelva un `char arbol2` donde cada nodo contenga únicamente la primer letra de la palabra del nodo correspondiente en el árbol de entrada. Si `firstLetter` tomara como entrada el árbol del ejercicio 1 el resultado esperado sería:



3. Resuelva los siguientes items usando las funciones `map`, `foldl`, `foldr` y `filter` o una combinación de ellas.
- (1 punto) Defina la función `listNeg` que tome una lista de reales y devuelva la lista de los opuestos a sus valores absolutos. *Ejemplo:* `listNeg([1,0,~7,4,~9]) = [~1,0,~7,~4,~9]`
 - (1 punto) Defina la función `prodMayores5` que tome una lista reales y devuelva el producto de aquellos valores mayores a 5. *Ejemplo:* `prodMayores5([~1,0,7,5,4,9]) = 63`

4. Considere la siguiente definición de árbol binario ya vista en clase y un nuevo tipo de dato **paciente**:

```
datatype 'etiqueta arbolbin =  
    Vacio |  
    Nodo of 'etiqueta * 'etiqueta arbolbin * 'etiqueta arbolbin;  
  
type paciente = {Nombre: string, Edad: int, Enfermedad: string, Medico: string };
```

Suponga que un hospital guarda el registro de todos sus pacientes en un árbol binario de búsqueda de tipo **paciente arbolbin** (no hay dos pacientes que se llamen igual). Cada nodo del árbol es una estructura que contiene los datos de un paciente (el nombre escrito todo en minúscula), su edad, enfermedad diagnosticada por la cual ha sido ingresado al hospital y su médico tratante. El árbol binario de búsqueda está ordenado de acuerdo al orden alfabético del campo **Nombre**. Implemente las siguientes funcionalidades que permitan realizar modificaciones/consultas sobre dicho registro:

- a. **(0.5 puntos)** Defina una función **ingresar_paciente** que tome el registro de todos los pacientes del hospital (implementada como un árbol binario de búsqueda) y los datos de un paciente y si dicha persona no se encuentra ya en el registro se agrega al mismo (recuerde mantener la propiedad de árbol binario de búsqueda). Si la persona ya estaba en el registro deben actualizarse sus datos.
Ayuda: Para esto defina antes una función **crear_paciente** que dados los datos de un paciente, cree la estructura paciente con dichos datos.
- b. **(1 punto)** Defina una función **buscar_pacientes_medico** que tome el registro de pacientes (implementado como un árbol binario de búsqueda) y un médico y devuelva la lista de pacientes (**paciente list**) que están siendo tratados por dicho médico. Si no hay pacientes tratados por ese médico debe devolver la lista vacía. Debe usar la función de alto orden **filter** para su definición.
- c. **(0.5 puntos)** Defina una función **actualizar_diag_enfermedad** que tome el registro de pacientes (implementado como un árbol binario de búsqueda), un nombre y una enfermedad y, si dicha persona se encuentra en el registro, actualice el registro del paciente con la nueva enfermedad diagnosticada. En caso de no encontrar la persona en el registro no debe modificar nada en el mismo.
- d. **(1.5 puntos)** Defina una función **buscar_nombres_edad** que tome el registro de pacientes (implementado como un árbol binario de búsqueda) y una edad mínima y una edad máxima y devuelva la lista de nombres (**string list**) de pacientes que están en el registro y cuyas edades pertenecen al rango definido por las edades dadas (puede que no haya ningún paciente). Debe usar una función de recorrido de árboles, la expresión **let** y alguna función de alto orden para su definición.
- e. **(1.5 puntos)** Defina una función **contar_pacientes_enfermedad** que tome el registro del hospital (implementado como un árbol binario de búsqueda) y una enfermedad y devuelva la cantidad de pacientes a los cuales se les diagnosticó dicha enfermedad. Debe usar una función de recorrido de árboles y alguna función de alto orden para su definición.