

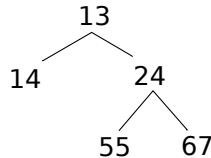


Nombre y Apellido:

Mail:

## Prueba

1. Dado el siguiente árbol:



representélo usando:

- a. el siguiente tipo de dato:
- ```
- datatype int arbolbin =  
    Vacio |  
    Nodo of int * int arbolbin * int arbolbin;
```
- b. el siguiente tipo de dato:
- ```
- datatype int arbol =  
    Nodo of int * int arbol list;
```

2. Considere la siguiente definición de árbol binario ya vista en clase:

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

Suponga que existe una agenda telefónica implementada como un árbol binario de búsqueda de tipo `(string, string, int) arbolbin`. Cada nodo del árbol es una terna que contiene el nombre de una persona (escrito todo en minúscula), la dirección y el teléfono. El árbol binario de búsqueda está ordenado de acuerdo al orden alfabético del campo nombre. Implemente las siguientes funcionalidades que permitan utilizar la agenda:

- a. Defina una función **ingresar-contacto** que tome una agenda (implementada como un árbol binario de búsqueda) y los datos de una persona (nombre, dirección y teléfono) y si dicha persona no se encuentra ya en la agenda se agrega a la misma (recuerde mantener la propiedad de árbol binario de búsqueda). Si la persona ya se encuentra en la agenda no debe modificar la agenda.
- b. Defina una función **modificar-contacto** que tome una agenda (implementada como un árbol binario de búsqueda) y los datos de una persona (nombre, dirección y teléfono) y si dicha persona se encuentra ya en la agenda modifica los datos asociados. Si la persona no se encuentra en la agenda no debe modificar la agenda.
- c. Defina una función **buscar-direccion** que tome una agenda (implementada como un árbol binario de búsqueda) y el nombre de una persona y devuelva su dirección en caso de que dicha persona esté en la agenda. Si la persona no se encuentra en la agenda debe devolver el mensaje "No se encontró el contacto buscado".
- d. Defina una función **buscar-telefono** que tome una agenda (implementada como un árbol binario de búsqueda) y el nombre de una persona y devuelva su teléfono en caso de que dicha persona esté en la agenda. Si la persona no se encuentra en la agenda debe devolver  $\sim 1$ .
3. Dado el siguiente tipo de dato que nos permite representar cantidades de dinero en pesos, euros y dólares:
- ```
- datatype Dinero = Euros of real | Pesos of real | Dolares of real
```
- y conociendo las siguientes equivalencias:

- 1 Peso = 15 Dólares
- 1 Peso = 17 Euros

Defina las siguientes funciones que tomen una entrada de tipo de dato Dinero y devuelvan un resultado de tipo Dinero según corresponda:

- a. Defina la función `convertir_a_euros` que tome un valor de tipo Dinero y devuelva su equivalente en euros, teniendo en cuenta las equivalencias propuestas más arriba.
- b. Defina la función `convertir_a_pesos` que tome un valor de tipo Dinero y devuelva su equivalente en pesos, teniendo en cuenta las equivalencias propuestas más arriba.
- c. Defina la función `convertir_a_dolares` que tome un valor de tipo Dinero y devuelva su equivalente en dólares, teniendo en cuenta las equivalencias propuestas más arriba.