```
--
--  Naive implementation of basic data base functionality
--
--  Author: Gabriele Keller
--
--  Remarks: elements have to be of type class Ord!
--           implementation is incomplete, various operations
--           missing

module Database (
  Database,
  newDB,          -- Ord a => Database a
                  -- return an empty database
  insertItemDB,   -- Ord a => Database a -> a -> Database a
                  -- insert item into database
  deleteItemDB,   -- Ord a => Database a -> a  -> Database a
                  -- delete item from database. If item not
                  -- in Database, return orig. Database
  changeItemDB,   -- Ord a => Database a -> a  -> a -> Database a
                  -- update item
  searchDB        -- Ord a => Database a -> (a -> Bool) -> [a]
                  -- 'search db p' returns all entries e of the
                  --  data base for which 'p e' evaluates to 'True'
) where


type Database a = [a]

newDB:: Ord a => Database a
newDB = []

-- insert a new item into database. Assumes items are
-- ordered, resulting list is also ordered
insertItemDB:: Ord a => Database a -> a -> Database a
insertItemDB []     item = [item]
insertItemDB (d:db) item
  | d < item  = (d : (insertItemDB db item))
  | otherwise =   d: (insertItemDB db item)

-- find all items in data base for which predicate
-- yields 'True'
searchDB:: Ord a => Database a -> (a -> Bool) -> [a]
searchDB db searchCrit =
    [item | item <- db, searchCrit item]


-- assumes list sorted. Search through list until either
-- item is found or head of list smaller than item
deleteItemDB:: Ord a => Database a -> a  -> Database a
deleteItemDB []     item = []
deleteItemDB (d:db) item
  | item == d = db
  | item < d  = (d:db)
  | otherwise = d : (deleteItemDB db item)



changeItemDB:: Ord a => Database a -> a  -> a -> Database a
changeItemDB []     oldItem newItem = []
changeItemDB (d:db) oldItem newItem
  | oldItem == d = (newItem: db)
  | oldItem < d  = (d:db)
  | otherwise = d : (changeItemDB db oldItem newItem)
```