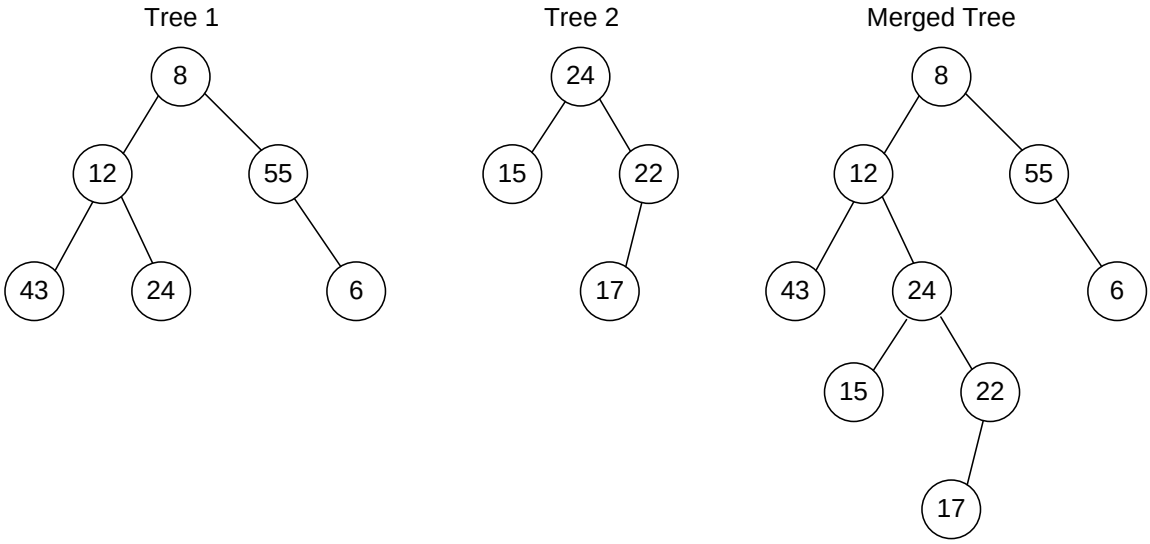
CENG 242

Hw #1

Spring 2007/2008

(Due: March 16th, 2008 Sunday 23:59)

You are given a list of binary trees. All the nodes of trees are numbered with an integer. You can merge two trees if the **root** of one has the same number with one of the **leaf**s of the other. A sample figure is given below.



You can merge *Tree 1* and *Tree 2* and get *Merged Tree* as shown in the figure. You should write a **Haskell** code merging the given list of trees. The tree is defined as follows:

```
data Tree = Empty | Branch Integer Tree Tree
```

You will write a function **mergeTrees** in the form:

```
mergeTrees::[Tree]->Tree
```

Constraints:

- A number will appear at most twice in the given list of trees and a number will appear at most once in a tree.
- If there are more than one solutions, both will be considered OK.
- If all of the trees in the given list can be merged to a single tree, you should give the **merged tree** as a result. Otherwise, you should return **Empty**.
- All the trees in the list will be nonempty.

Examples:

- `mergeTrees [Branch 8 (Branch 12 (Branch 43 Empty Empty) (Branch 24 Empty Empty)) (Branch 55 Empty (Branch 6 Empty Empty)), Branch 24 (Branch 15 Empty Empty) (Branch 22 (Branch 17 Empty Empty) Empty)]`

This is the example given in the figure and the result is:

(Branch 8 (Branch 12 (Branch 43 Empty Empty) (Branch 24 (Branch 15 Empty Empty) (Branch 22 (Branch 17 Empty Empty) Empty))) (Branch 55 Empty (Branch 6 Empty Empty)))

- `mergeTrees [Branch 6 (Branch 7 Empty Empty) Empty, Branch 5 (Branch 6 Empty Empty) Empty, Branch 7 Empty (Branch 8 Empty Empty)]`

(Branch 5 (Branch 6 (Branch 7 Empty (Branch 8 Empty Empty)) Empty) Empty)

- `mergeTrees [Branch 5 Empty (Branch 10 Empty Empty), Branch 9 (Branch 3 Empty Empty) (Branch 5 (Branch 8 Empty Empty) Empty), Branch 10 Empty (Branch 33 Empty Empty)]`

You can merge first and third trees but second tree can not be merged with any other. So the result is:

Empty

Specifications:

- All the work should be done **individually**.
- Your codes should be written in **Haskell** and have the name "**Hw1.hs".**
- Your code should have the module **Hw1** (i.e. your code should start with: **module Hw1 where**)
- In evaluation, black box method will be used. So, be careful about names of functions, data structures etc.
- You will submit your codes through **cow** system.
- You should test your codes in **inek** machines with **hugs** before submitting.\

Note:

You can not view a **Tree** in interpreter if you do not implement a **show** function for it. So use this function in order to see your trees:

```
showTree :: Tree -> String
showTree Empty = "Empty"
showTree (Branch x left right) = "(Branch " ++ show x ++ " " ++ showTree left
++ " " ++ showTree right ++ ")"
```