

QUESTION 4

```
class Trie {
```

```
  private val children = new Array [Trie] (26) // children of this node
  private var complete = false // does a word end at this node?
```

```
  private def toIndex (c: Char): Int = c.toInt - 'a'.toInt
```

```
  private def toChar (i: Int): Char = (i + 'a'.toInt).toChar
```

```
// (a)
```

```
def contains (w: String): Boolean = {
```

```
  var i = toIndex(w(0))
```

```
  if (w.size == 1) // we got to the last letter of the word
```

```
  { if ((children(i) != null) && (children(i).complete == true)) return true
    else return false
```

```
  }
```

```
  else
```

```
  { if (children(i) != null) return children(i).contains(w.tail) // use recursion
    else return false
```

```
  }
```

```
}
```

```
// (b)
```

```
def add (w: String): Unit = {
```

```
  var i = toIndex(w(0))
```

```
  if (w.size == 1)
```

```
  { if (children(i) == null) children(i) = new Trie
    children(i).complete = true
```

```
  }
```

```
  else
```

```
  { if (children(i) == null) children(i) = new Trie
    children(i).add(w.tail)
```

```
  }
```

```
}
```

// (c)

```
def allWords (ls: String): List[String] = {  
  var cnt = new Array[Int](26)  
  for (c <- ls) cnt(toIndex(c)) += 1  
  run(this, cnt, "")  
}
```

```
private def run (node: Trie, cnt: Array[Int], pref: String): List[String] = {  
  var answer = List[String]()  
  if (node.complete) answer = List[String](pref)  
  for (i <- 0 until 26) {  
    if (cnt(i) > 0 && node.children(i) != null) {  
      cnt(i) -= 1  
      answer ++= run (node.children(i), cnt, pref + toChar(i))  
      cnt(i) += 1  
    }  
  }  
  answer  
}
```

object Test {

```
  def include (w: String): Boolean = w.forall(_._isLower)
```

```
  def main (args: Array[String]) = {
```

```
    var head = new Trie
```

```
    val lines = scala.io.Source.fromFile(args(0)).getLines
```

```
    for (w <- lines)
```

```
      if (include(w)) head.add(w)
```

```
    println (head.allWords(args(1))
```

```
  }
```

```
}
```