You may find it useful before implementing some of the problems in the Programming Assignment to look closer at the pseudocode for the algorithms discussed in the lectures.

Here is the pseudocode for constructing a trie from a collection of patterns:

```
TrieConstruction(Patterns)
    Trie ← a graph consisting of a single node root
    for each string Pattern in Patterns
        currentNode ← root
        for i ← 1 to |Pattern|
            currentSymbol ← i-th symbol of Pattern
            if there is an outgoing edge from currentNode with label currentSymbol
                currentNode ← ending node of this edge
            else
                add a new node newNode to Trie
                add a new edge from currentNode to newNode with label currentSymbol
                currentNode ← newNode
    return Trie
```

Here is the pseudocode for matching a collection of patterns against the text using a trie:

```
PrefixTrieMatching(Text, Trie)
    symbol ← first letter of Text
    v ← root of Trie
    while forever
        if v is a leaf in Trie
            return the pattern spelled by the path from the root to v
        else if there is an edge (v, w) in Trie labeled by symbol
            symbol ← next letter of Text
            v ← w
        else
            output "no matches found"
            return
TrieMatching(Text, Trie)
    while Text is nonempty
        PrefixTrieMatching(Text, Trie)
        remove first symbol from Text
```