```
Bool read_sym(int infile, uint8_t *sym){
	Static uint8_t chars[size];
	Static uint16_t index = 0;
	Static uint16_t end = 0;

	If index == 0
		End = read
	If end == 0
		Return false
	If index < end
		*sym = chars[index]
		Index += 1
		Return true
	Else
		Index = 0

	Return true;
}

Trienode trie_node_create(uint16_t code){
	Trienode trienode = malloc sizeof trienode
	If trienode
		trienode->code = code
		For int i = 0; i < alphabet; i++
			trienode->children[i] = NULL;
		Return trienode
	Else
		Print error

	Return null
}

Void trie node delete(trienode n){
	If n
		Free n
		N = null
}

Trienode trie create{
	Trienode node = trie node create(empty code);
	Return node
}
```

```
Void trie reset(trienode root){
        For int i = 0; i < alphabet; i++
                If root->children[i] != null
                        trie_delete(root->children[i])
}

Void trie delete(trienode n){
        For int i = 0; i < alphabet; i++
                If n->children[i] != null
                        Trie delete(n->children[i])
        Trie node delete(n)
}

Trienode trie_step(trienode n, sym){
        If n->children[sym] != null
                Return n->children[sym]
        Return null
}
```