Appendix B: Parser test programs

```
./parse.js
//import * as fs from 'node:fs';
//import { parse } from  "../jison/occam.js";

const fs = require('fs');
const occam = require("../jison/occam").parser;

const testfilepath = "/home/august/boccam/test/parsefiles/"
var files = fs.readdirSync(testfilepath);

for (const f of files) {
    console.log(f)
    console.log("    " + testParsing(f))
    console.log("")
}

function testParsing (filename) {

    const file = fs.readFileSync(testfilepath + filename, 'utf-8')
    const asserts = file.split("---")[1].split('\n')
    const code = file.split("---")[2] + " "

    var output = "SUCCESS"

    try {
        occam.parse(code);
    } catch (e) {
            try {
                output = "Parsing Error: " + e.toString();
            } catch (e2) {
                throw e
            }
    }

    if (asserts.includes("parse: true")) {
        if (output != "SUCCESS") {
            output = "TEST FAILED: Successful parse expected, got " + output
        }
    } else {
        if (output == "SUCCESS") {
            output = "TEST FAILED: Failed parse expected"
        }
    }

    return output
}
./notwrappedalt.occ
```

```
---
parse: false
---
CHAN OF INT myChan:
INT x:
PAR
    SEQ i = [0 FOR 31]
        GRAPHICS[i][0] ! 3
        myChan ! 1
    SEQ i = [0 FOR 31]
        SEQ
            GRAPHICS[i][0] ! 7
            myChan ? x
./seqdepth.occ
---
parse: false
---
SEQ
    CHAN OF INT myChan:
      INT x:
./pardepth.occ
---
parse: false
---
PAR
    CHAN OF INT myChan:
      INT x:
./acrosslines.occ
---
parse: false
---
SEQ
    CHAN OF INT myChan:
    [5]CHAN OF INT x:
    SEQ i = [0 FOR 31]
        SEQ
            myChan ? x
            [3] ! 0
./indentdepth.occ
---
parse: true
---
SEQ
    INT x:
    WHILE TRUE
      x := (x+1)
./acrosslines2.occ
---
parse: false
```

```
---
SEQ
    [5]INT x:
    x
      [3] := 1
./sameline.occ
---
parse: false
---
SEQ
    INT x:
    x := 1
    CHAN OF INT chan:
    CHAN OF INT chan2:
    PAR
        SEQ
            chan ! x
            SERIAL ! 0
        SEQ
            chan2 ? x SERIAL ! 1
        SERIAL ! 2

./notwrapped.occ
---
parse: false
---
SEQ
    CHAN OF INT myChan:
    INT x:
    PAR
        SEQ i = [0 FOR 31]
            SEQ
                GRAPHICS[i][0] ! 3
                myChan ! 1
        SEQ i = [0 FOR 31]
            SEQ
                GRAPHICS[i][0] ! 7
                myChan ? x
```