TSR – Lab 1

This exam consists of 10 multiple choice questions. In every case only one answer is correct. You should answer in a separate sheet. If correctly answered, they contribute 1 point to the exam grade. If incorrectly answered, the contribution is negative: -0.333. So, think carefully your answers.

This is the “exam.js” program to be considered in the rest of this exam (You can dettach next page from the others. Thus, you’ll have program and questions side by side):

```
var fs = require('fs');
var net = require('net');

var functions = []

module.exports = function (what, how) {
  functions[what](how);
}
// ===========================================
functions.push(
  function (how) {
    fs.readFile(how, 'utf8', function(err,data) {
      console.log(err);
    });
  }
)
functions.push(
  function (how) {
    fs.readFile(how, 'utf8', console.log);
  }
)
functions.push(
  function (how) {
    var data = fs.readFileSync(how, 'utf8');
    console.log (data);
  }
)
functions.push(
  function (n) {
    function loop (n) {
      var f = 0;
      for (var i = 0; i <= n; i++) f = i;
      return f;
    }
    setTimeout(function() {
      console.log("TIMEOUT")}, 1000);
    console.log(loop(n/2));
    console.log(loop(n));
  }
)
```

```
functions.push(
  function (how) {
    fs.readFile(how, 'utf8', function (data) {
      how = 99;
    });
    console.log (how);
  }
)
functions.push(
  function (how) {
    net.connect(JSON.parse(how), function () {
      console.log ("CONNECTED");})
  }
)
functions.push(
  function (how) {
    net.connect(how, function (err) {
      if (err) console.log ("NOT CONNECTED")})
  }
)
functions.push(
  function (how) {
    var s1 = net.createServer(function (c) {
      console.log ("CONNECTED 1")
    });
    var s2 = net.createServer(function (c) {
      console.log ("CONNECTED 2")
    });
    s1.listen (how);
    s2.listen (how*1 + 1);
    console.log ("READY");
  }
)
functions.push(
  function (how) {
    net.connect({port:how});
  }
)
```

Note that “functions” is an array of functions. Initially, it has been handled as a queue in order to insert in it each one of those functions.

Suppose you start a NodeJS interactive console on the same directory you have the above file. Assuming you execute the following command:

```
node doexam.js X Y
```

where the contents of `doexam.js` are:

```
var myFuncs=require(’./exam.js’);
myFuncs(process.argv[2],process.argv[3]);
```

Answer the questions about the output of the above console command for various values of X and Y, under a variety of conditions.

TSR – Lab 1

1. When X = 0, Y = '/tmp/hosts', and "/tmp/hosts" does not exist. It prints…

a	Nothing
b	"undefined"
c	"null"
d	A representation of an error condition returned by *readFile*

2. When X = 0, Y = '/tmp/hosts', and "/tmp/hosts" exists. It prints…

a	The contents of the /tmp/hosts file, as a string coded in *utf8*
b	Nothing. It throws an exception.
c	"null"
d	A representation of an error condition returned by *readFile*

3. When X = 1, Y = '/tmp/hosts', and "/tmp/hosts" does not exist. It prints…

a	Nothing, it throws an exception instead
b	Nothing, but exits silently
c	A representation of an Error condition returned by *readFile*
d	"undefined"

4. When X = 1, Y = 'exam.js'. It prints…

a	The contents of the functions array, in JSON format.
b	"null" followed by the contents of "exam.js".
c	A representation of an Error condition returned by *readFileSync*
d	"undefined"

5. When X = 3, Y = 10000, assuming each loop pass takes 1ms, it prints…

a	5000, then TIMEOUT, then 10000
b	TIMEOUT, then 5000, then 10000
c	10000, then 5000, then TIMEOUT
d	5000, then 10000, then TIMEOUT

TSR – Lab 1

6. **When X = 4, Y = 33, it prints...**

a	33
b	99
c	Nothing
d	A representation of an error

7. **When X = 5, Y = ’{“port”: 80, “host”: “www.upv.es”}’, and a process is listening on port 80 of host ‘www.upv.es’, it prints...**

a	Nothing, throws an exception
b	CONNECTED
c	A representation of an error object
d	The web page of www.upv.es in textual form (as HTML code)

8. **When X = 5, Y = ’{“port”: 80, “host”: “nonexistent.address”}’, it prints...**

a	It throws an exception and the process is aborted. Nothing else is printed.
b	CONNECTED
c	Nothing, and the process ends.
d	An HTTP error code.

9. **When X = 7, Y = 8000, it prints...**

a	Nothing, throws an exception as we cannot bind to two different ports within a NodeJS process (single-threaded)
b	“CONNECTED 1”
c	“CONNECTED 2”
d	“READY”

10. **When X = 7, Y = 8000, and, in another terminal launched on the same directory we execute node doexam.js 8 8001, the original terminal prints...**

a	Nothing, throws an exception as we cannot bind to two different ports within a nodejs process (single-threaded)
b	“READY”, and then “CONNECTED 2”
c	“CONNECTED 2” and then “READY”
d	“CONNECTED 2”