TSR – Pràctica 1

Aquest examen consta de 10 qüestions d'opció múltiple. En cada qüestió solament una de les respostes és correcta. Has de contestar en una fulla a part. Les respostes correctes aporten 1 punt a la teua qualificació. Les incorrectes descompten 0.333 punts.

Aquest és el programa "exam.js" a utilitzar en la resta de l'examen. Pots soltar la següent fulla de les altres. Així, podràs tenir el programa i les qüestions visibles simultàniament:

```
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});
  }
)
```

Observa que "functions" és un vector de funcions, gestionat en la seua iniciació com una cua on s'han anat inserint les funcions.

Suposa que s'inicia una consola NodeJS interactiva en el directori on es troba aquest fitxer. Imagina que s'executa aquesta ordre en la consola:

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

sent el contingut de `doexam.js` el següent:

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

Respon les següents qüestions sobre l'eixida proporcionada per aquesta ordre per a diferents valors de X i Y, en les condicions que es descriuen.

TSR – Pràctica 1

1. Quan X = 0, Y = '/tmp/hosts', i "/tmp/hosts" no existisca, mostrarà...

a	No res.
b	"undefined"
c	"null"
d	Una representació d'una condició d'error, generada per *readFile.*

2. Quan X = 0, Y = '/tmp/hosts', i "/tmp/hosts" existisca, mostrarà…

a	El contingut del fitxer /tmp/hosts, com una cadena codificada en UTF8.
b	No res. Es genera una excepció.
c	"null"
d	Una representació d'una condició d'error, generada per *readFile.*

3. Quan X = 1, Y = '/tmp/hosts', i "/tmp/hosts" no existisca, mostrarà...

a	No res. Es generarà una excepció.
b	No res, però finalitza sense errors ni excepcions.
c	Una representació d'una condició d'error, generada per *readFile.*
d	"undefined"

4. Quan X = 1, Y = 'exam.js', mostrarà...

a	El contingut del vector de funcions representat en format JSON.
b	"null" seguit del contingut d'exam.js.
c	Una representació d'una condició d'error, generada per *readFile.*
d	"undefined"

5. Quan X = 3, Y = 10000, assumint que cada iteració necessita 1 ms, mostrarà...

a	5000, seguit de TIMEOUT i després 10000.
b	TIMEOUT, seguit de 5000 i després 10000.
c	10000, seguit de 5000 i després TIMEOUT.
d	5000, seguit de 10000 i després TIMEOUT.

TSR – Pràctica 1

6. Quan X = 4 i Y = 33, mostrarà...

a	33
b	99
c	No res.
d	Un error.

7. Quan X = 5, Y = '{"port": 80, "host": "www.upv.es"}', i un procés atenga en el port 80 de 'www.upv.es', imprimirà...

a	No res. Generarà una excepció.
b	CONNECTED
c	Una representació d'un objecte d'error.
d	La pàgina web de www.upv.es en format text (com a codi HTML).

8. Quan X = 5, Y = '{"port": 80, "host": "nonexistent.address"}', imprimirà...

a	Generarà una excepció que avorta el procés.
b	CONNECTED
c	No res, i el procés acaba.
d	Un codi d'error HTTP.

9. Quan X = 7, Y = 8000, imprimirà...

a	No res. Generarà una excepció perquè no es pot fer bind() sobre dos ports diferents en un procés NodeJS (ja que només té un fil d'execució).
b	"CONNECTED 1"
c	"CONNECTED 2"
d	"READY"

10. Quan X = 7, Y = 8000, i en una altra terminal en el mateix directori s'execute node doexam.js 8 8001, la terminal original mostrarà...

a	No res. Generarà una excepció perquè no es pot fer bind() sobre dos ports diferents en un procés NodeJS (ja que només té un fil d'execució).
b	"READY" i després "CONNECTED 2"
c	"CONNECTED 2" i després "READY"
d	"CONNECTED 2"