TSR – Práctica 1

Este examen consta de 10 cuestiones de opción múltiple. En cada cuestión solo una de las respuestas es correcta. Debes contestar en una hoja aparte. Las respuestas correctas aportan 1 punto a tu calificación. Las incorrectas descuentan 0.333 puntos.

Este es el programa "exam.js" a utilizar en el resto del examen (Puedes soltar la última hoja de las demás. Así, podrás tener el programa y las cuestiones visibles simultáneamente):

```
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" es un vector de funciones, manejado en su creación como una cola para ir insertando en él cada una de las funciones.

Supón que se inicia una consola NodeJS interactiva en el directorio donde se encuentra este fichero. Imagina que se ejecuta esta orden en esa consola:

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

siendo el contenido de `doexam.js` el siguiente:

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

Responde las siguientes cuestiones sobre la salida proporcionada por esa orden para diferentes valores de X e Y, bajo las condiciones que se describen.

TSR – Práctica 1

1. Cuando X = 0, Y = '/tmp/hosts', y "/tmp/hosts" no exista, mostrará...

a	Nada
b	"undefined"
c	"null"
d	Una representación de una condición de error, generada por *readFile.*

2. Cuando X = 0, Y = '/tmp/hosts', y "/tmp/hosts" exista, mostrará…

a	El contenido del fichero /tmp/hosts, como una cadena codificada en UTF8.
b	Nada. Se generará una excepción.
c	"null"
d	Una representación de una condición de error, generada por *readFile.*

3. Cuando X = 1, Y = '/tmp/hosts', y "/tmp/hosts" no exista, mostrará...

a	Nada. Se generará una excepción.
b	Nada, pero finaliza sin errores ni excepciones.
c	Una representación de una condición de error, generada por *readFile.*
d	"undefined"

4. Cuando X = 1, Y = 'exam.js', mostrará...

a	El contenido del vector de funciones representado en formato JSON.
b	"null" seguido del contenido del fichero "exam.js".
c	Una representación de una condición de error, generada por *readFile.*
d	"undefined"

5. Cuando X = 3, Y = 10000, asumiendo que cada iteración necesita 1 ms, mostrará...

a	5000, seguido de TIMEOUT y luego 10000.
b	TIMEOUT, seguido de 5000 y luego 10000.
c	10000, seguido de 5000 y luego TIMEOUT.
d	5000, seguido de 10000 y luego TIMEOUT.

TSR – Práctica 1

6. Cuando X = 4 e Y = 33, mostrará...

a	33
b	99
c	Nada
d	Un error.

7. Cuando X = 5, Y = '{"port": 80, "host": "www.upv.es"}', y un proceso atienda en el puerto 80 de 'www.upv.es', imprimirá...

a	Nada. Generará una excepción.
b	CONNECTED
c	Una representación de un objeto de error.
d	La página web de www.upv.es en formato texto (como código HTML).

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

a	Generará una excepción que aborta el proceso.
b	CONNECTED
c	Nada, y el proceso termina.
d	Un código de error HTTP.

9. Cuando X = 7, Y = 8000, imprimirá...

a	Nada. Generará una excepción porque no se puede hacer bind() sobre dos puertos distintos en un proceso NodeJS (pues solo tiene un hilo de ejecución).
b	"CONNECTED 1"
c	"CONNECTED 2"
d	"READY"

10. Cuando X = 7, Y = 8000, y en otra terminal en ese mismo directorio se ejecute node doexam.js 8 8001, la terminal original mostrará...

a	Nada. Generará una excepción porque no se puede hacer bind() sobre dos puertos distintos en un proceso NodeJS (pues solo tiene un hilo de ejecución).
b	"READY" y después "CONNECTED 2"
c	"CONNECTED 2" y después "READY"
d	"CONNECTED 2"