



1 Linguagens de Programação 1 (30 Pontos)

- (i) Explique brevemente a diferença entre uma linguagem compilada e uma puramente interpretada.
- (ii) Explique o conceito de “scope” numa variável em JavaScript. Usar como exemplo as instruções
 $x = 2$, `let x=2` e `var x=2`
- (iii) Explique a diferença entre tipificação forte e fraco em relação às linguagens Java, JavaScript e Bash.

2 Bash Shell Scripts (40 pontos)

Considere o ficheiro “/etc/passwd” de Linux cujas linhas são do tipo

“username:password-info:user-id:group-id:descrição:directório raiz:interpretador” .

por exemplo : `jsmith:x:1001:1000:Joe Smith,Gab 6.1,92342323:/home/jsmith:/bin/bash`

- Escreva um bash shell (ex2.sh) para (i) imprimir para o ecrã as linhas com utilizadores que tem como interpretador por defeito “bourne shell” (/bin/sh). O output deverá ser ordenado pelo user-id

3 JavaScript : Funções e Objetos (50 pontos)

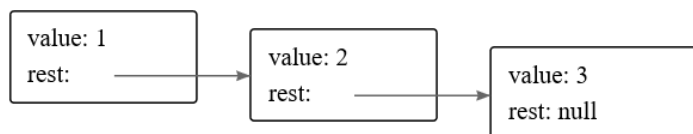
Considere o construtor para um objeto do tipo número complexo ($z = x + iy$). Um número complexo será representado usando dois valores, um para a parte real e o outro para a parte imaginária

```
function numeroComplexo(real, img) {  
  this.real = real;  
  this.img = img;  
}
```

- i. Acrescente a esta definição um método para imprimir os valores do objeto (utilize `console.log()`)
- ii. Acrescente a esta definição um método para calcular o seu módulo ($z = x + iy \rightarrow |z| = \sqrt{x^2 + y^2}$)
- iii. Escreva um pequeno script para declarar um Array de 100 objetos deste tipo inicializados com valores aleatórios (`Math.random()`), ordenar o Array pelo valor de módulo e para imprimir o objeto com módulo maior.

4 Objetos e Recursividade: JavaScript (30 pontos)

Em JavaScript podemos declarar objetos ‘literais’ com a estrutura duma lista ligada (uma estrutura de dados linear e dinâmica composta por células que apontam para o próximo elemento da lista). No exemplo que se segue uma lista é composta por dois atributos, value (um valor inteiro) e rest (um objeto).



```
var listaA = { value: 1, rest: { value: 2, rest: { value: 3, rest: null } } };
```

- Defina uma nova variável chamada *listaB* com os valores “4” e “5” e insere esta nova lista no fim da *listaA*
- Escreva uma função “recursiva” para contar o número de elementos duma lista deste tipo.

5 JavaScript - Node.js (50 pontos)

Considera a aplicação da framework Express seguinte no ficheiro “index.js”. As variáveis *cinemas* e *idades* são *arrays* onde um índice por exemplo representa o nome dum cinema e a sua cidade de localização.

```
var express = require('express');
var app = express();
let cinemas = [ "Cineplace-Serra", "Cineplace", "Cinebox" ]
let cidade = [ "Covilha", "Guarda", "Castelo-Branco"]
app.get('/', (req, res) => {
  var cidade = req.query.name;
  res.status(200).type('html');
  let inx = cinemas.indexOf( cidade );
  if ( -1 === inx )
    res.write ( "<h1>Cinema Nome Não Encontrado</h1>" )
  else
    res.write ( "<h1>Cinema Nome "+cinemas[inx]+"</h1>" );
  res.end();
} );
app.listen( 8088, () => { console.log('App em Execução') } );
```

- O que faz o comando `npm install` ? Que ficheiro altera ou lê ?
- Se a aplicação estiver iniciada com o comando “`node index.js`” numa máquina pessoal (localhost), qual seria a URL necessária para um cliente receber do servidor o nome do cinema na Guarda..
- Escreva o código necessário para adicionar uma função no endpoint (ponto de acesso) `/info` para a aplicação retornar dentro dum objeto JSON os dois vetores(arrays), *cinemas* e *idades*.
- Escreva o código necessário para adicionar uma função no endpoint (ponto de acesso) `/insert` para a aplicação inserir um novo nome dum cinema e reescreva cidade e descreva como se usava esta endpoint.