

Grupo 1 - Interpretação

Considere as funções misteriosas e constantes JavaScript (JS) definidas na última página do enunciado.

1. Qual o valor de q1 depois de executar-se o código seguinte?

```
var q1 = f1(FRASE1)
```

2. Qual o valor de q2 depois de executar-se o código seguinte?

```
var q2 = misterio1(f1.toString)
```

3. Qual o valor de q3 depois de executar-se o código seguinte?

```
var q3 = misterio1(FRASE1)
```

4. Qual o valor de q4 depois de executar-se o código seguinte?

```
var q4 = misterio2(FRASE2)
```

5. Qual o valor de q5 depois de executar-se o código seguinte?

```
var q5 = misterio3(COL1)
```

6. Qual o valor de q6 depois de executar-se o código seguinte?

```
var q6 = misterio4(COL2)
```

7. Escolha a opção que corresponda ao valor de q7 depois de executar-se o código seguinte.

```
q7 = misterio5(COL2, COL2.length, COL2.length>10)
```

A) 0

B) 1

C) 2

D) 3

E) nenhuma das anteriores

8. Qual o valor de q8 depois de executar-se o código seguinte?

```
var q8 = misterio6(3)
```

9. O que aparece escrito, ao executar-se o código seguinte?

```
document.write(misterio7())
```

10. O que aparece escrito, ao executar-se o código seguinte?

```
document.write(misterio8())
```

Grupo 2 - HTML + JS

Em todos os bilhetes de identidade (BI) e/ou Cartões de Cidadão (CC) de cidadãos de Portugal, à direita do número de cidadão, aparece um **"número de controlo"**, que é calculado por um algoritmo de segurança, explicado de seguida para um hipotético nº de BI/CC "123".

Atribuem-se "pesos", **a começar em 2**, aos dígitos do nº do BI/CC, **da direita para a esquerda**.

Assim, no número "123", indo da direita para a esquerda:

- o dígito mais à direita é o "3", por isso tem peso 2;
- depois tem-se o dígito "2", com peso 3;
- depois, finalmente, temos o dígito "1", com peso 4;

Estes "pesos" servem para calcular um valor importante, chamado a **"soma do BI/CC"**.

Nesta soma, **cada dígito é multiplicado pelo peso** correspondente e somam-se todas essas parcelas.

Assim, no número "123", as parcelas da "soma do BI/CC" são: $3*2 + 2*3 + 1*4 = 6 + 6 + 4 = 16$

Finalmente, o **"número de controlo"** é o **primeiro número inteiro em [0, 10] que, quando adicionado à "soma do BI/CC" dá um resultado múltiplo de 11** (resto de divisão inteira por 11 idêntico a zero).

No caso do número "123", cuja "soma do BI" é 16, itera-se de a 0 a 10, à procura do primeiro múltiplo de 11:

$16 + 0 = 16$ (não é múltiplo de 11) ... $16 + 1 = 17$ (não é múltiplo de 11) ...

$16 + 6 = 22$ (é múltiplo de 11, STOP!)

Assim, o "número de controlo" para o número de BI/CC "123" é 6, calculado conforme descrito.

Nesta questão deve escrever um Sistema Web HTML+JS que calcule a "soma do BI" e o "DC/Dígito de Controlo/Número de Controlo" do nº do BI/cartão de cidadão.

→ 2.1 Escreva o HTML suficiente para conseguir uma form como a sugerida pela imagem à direita.

Os elementos relevantes têm os seguintes "id":

"idForm" - a form;

"idBI" - o campo onde está escrito "123";

"idDC" - o campo para escrita do DC;

"idSectionFeedback" - uma zona onde aparecerá feedback, inicialmente vazia;

- um recurso de "submit".

Nota: na imagem, abaixo da form, aparecem escritas algumas informações, apenas para ilustrar e deixar claro que existe um elemento de HTML naquele sítio.

Dados do BI

Um nº de BI/cartão do cidadão:

Digito de controlo:

submeter dados

Soma do BI = 16

DC = 6

→ 2.2 Escreva a function com assinatura

function numeroControlo (pBI)

que, recebida uma frase representativa de um número de BI, retorna o seu "número de controlo", calculado conforme explicado no início da questão 2.

→ 2.3 Escreva, utilizando o padrão de desenvolvimento "boot" (HTML+JS) que aplicou em aula, todo o JS necessário para conseguir o seguinte comportamento do sistema: quando se faz submit da form, deve escrever-se, na zona de feedback, uma informação de "aceite" ou de "rejeitado", consoante o DC esteja de acordo, ou não, com o BI/CC. Assuma que a função JS **numeroControlo** esta disponível e perfeita.

Grupo 3 - Python/Flask

Em alternativa a fazer-se a validação do número de BI/CC com JS, poderia utilizar-se uma app Python/Flask para esse efeito.

→ 3.1 Escreva uma solução Python/Flask que possa ser chamada por POST na route "/check", que possa substituir a versão JS atual, na verificação de um "número de BI/CC" relativamente a um "número de controlo".

→ 3.2 Como articularia essa app com o HTML+JS, para obter o comportamento pedido na questão 2.3?

```

const FRASE1 = "23-05-24"
const FRASE2 = "okidoki"
const COL1 = ["tudo", "vai", "correr" , "bem"]
const COL2 = [44, 333, 11]

function f1(){"Hello "}
f1.toString="oi"

function f2(pNome){return f1()+pNome+"!"}

function misterio0(pNome){return f2(pNome)}

function misterio1(pFrase){
  var ret = ""
  for(
    var idx=pFrase.length-1;
    idx>0;
    idx-=1
  ){
    ret+=pFrase.charAt(idx)
  }//for
  return ret
}//misterio1

function misterio2(pFrase){
  var ret = {}
  for(var idx=0; idx<pFrase.length; idx+=1)
    ret["p"+pFrase[idx]]=pFrase[idx]
  return ret.pk
}//misterio2

function misterio3(pCol){
  pCol.reverse()
  return pCol.unshift()
}//misterio3

```

```

function misterio4(pCol){
  if(pCol.length>=2)
    return pCol[1]>pCol[0] ? pCol[2] : pCol[0]
  else
    return pCol[0]
}//misterio4

function misterio5(pA, pB, pC=false){
  pA = pA[0]
  return !pC ? pA % (pB-1) : pB * pA
}//misterio5

function misterio6(p) {
  return p <= 0 ? -1 : misterio6(p - 1) + (2*p)
}//misterio6

function misterio7(pLimit=1){
  var strHtml = "<table border='1'>"
  for(var x=0; x<=pLimit; x+=1){
    var vx=x===0?true:false
    strHtml+="<tr>"
    for(var y=0; y<=pLimit; y+=1){
      var vy=y===0?false:true
      strHtml+=`<td>${vx}</td><td>${vy}</td>`
      strHtml+=`<td>${!vx || vy}</td>`
      strHtml+="</tr>"
    }//for
  }//for
  strHtml+="</table>"
  this.length=strHtml.length
  return strHtml
}//misterio7

function misterio8(p){
  return typeof((new misterio7()).length)
}//misterio8

```