



## **CURSO PROFISSIONAL TÉCNICO DE GPSI**

### **PROGRAMAÇÃO E SISTEMAS DE INFORMAÇÃO**

#### **MÓDULO 4 – ESTRUTURAS DE DADOS ESTÁTICAS**

#### **TESTE DE AVALIAÇÃO PRÁTICA**

---

Crie uma pasta no ambiente de trabalho cujo nome deve identificar a turma, número e nome. Por exemplo **10H\_01\_AnaPeres**. O nome do ficheiro deve identificar o número do problema. Por exemplo **PROBLEMA\_01.PY**. Guarde os ficheiros na pen do professor.

## Problema 1

---

### **Introdução**

Numa corrida de automóveis, participaram 5 pilotos com o respetivo veículo. No início da prova, cada piloto deve registar o seu nome.

### **Problema**

Elabore um programa que leia o nome do piloto e o tempo gasto na corrida em minutos. No final, determina e imprime o nome dos pilotos que cortaram a meta em primeiro e último lugar e a diferença entre esses dois tempos. O programa deve também listar o nome e o tempo gasto na corrida de TODOS os participantes.

### **Dados de entrada**

Os dados de entrada devem ser guardados em dois vetores (um para o nome e outro para o tempo gasto na corrida).

Os dados são escritos numa só linha separados por uma vírgula e um espaço (“; “).

Os vetores também podem ser inicializados no código de forma implícita.

Os dados referentes ao nome e tempo devem ser previamente guardados numa estrutura de dados.

### **Restrições**

O tempo gasto na corrida é um inteiro positivo.

### **Dados de saída**

Caso for encontrado tempos iguais, deve-se considerar o primeiro piloto guardado na estrutura de dados.

## **Exemplo**

### **Dados de entrada**

Nomes: Carlos Pimenta, Pedro Miguel, Maria, Simão, Paulo Reis

Tempos Gastos: 45, 48, 40, 41, 42

### **Dados de saída**

Primeiro lugar: Maria

Último lugar: Pedro Miguel

Diferença: 8 minutos

Nomes e tempos gastos na corrida:

Carlos Pimenta: 45 minutos

Pedro Miguel: 48 minutos

Maria: 40 minutos

Simão: 41 minutos

Paulo Reis: 42 minutos

## Problema 2

---

### **Introdução**

Uma das formas mais importantes de garantir a segurança das suas contas online é de criar uma palavra-passe segura

### **Problema**

Cria um programa que leia os dados de login (e-mail e password) e verifique se a palavra-passe introduzida é segura ou não. Para que a senha seja considerada segura deve-se obedecer às seguintes regras:

- a) Pelo menos 8 caracteres de comprimento;
- b) Combinação de letras maiúsculas, letras minúsculas e números;
- c) Não é uma palavra que possa ser encontrada no nome do utilizador de email (ex: se o email for serginho@esenviseu.net a password não pode ser “serginho”).

### **Dados de entrada**

Duas linhas de texto, uma para o email e outra para a password.

**Exemplo de dados de entrada 1**

e-mail: pedro@gmail.com

password: lindo+

**Exemplo de Dados de saída 1**

Senha não segura

**Exemplo de dados de entrada 2**

e-mail: pedro@gmail.com

password: MuitoLindo2

**Exemplo de Dados de saída 2**

Senha segura

## Cotações

### Problema1

Leitura dos dados de entrada .....	5
Registo do nome e dos tempos nos respetivos vetores .....	5
Validação dos dados de entrada .....	5
Calcular o nome do piloto que cortou a meta em primeiro lugar .....	15
Calcular o nome do piloto que cortou a meta em último lugar .....	15
Calcular a diferença entre esses dois tempos .....	10
Listar o nome e o tempo gasto na corrida de todos os participantes.....	20
Apresentação dos dados .....	10
Eficiência do programa.....	15

### Problema2

Leitura dos dados de entrada.....	10
Validação comprimento da senha .....	15
Validação letras em maiúsculas .....	10
Validação letras em minúsculas .....	10
Validação dígitos numéricos .....	10
Validação nome de email .....	15
Apresentação da resposta válida .....	10
Eficiência do programa .....	20
<b>TOTAL .....</b>	<b>200</b>