

Fundamentos de Programação

Avaliação Final 2

2015/2016

Objectivos:

- Programação em Python
- Estruturas de Dados
- Ordenação e Pesquisa

Duração

- Deverá completar os exercícios propostos em 2h00

Instruções

- Faça login com o username **sessao1** e a password **um**.
- No Desktop encontra quatro ficheiros (F2.py, pacientes.json, cores.json e sintomas.csv).
- Deve editar **F2.py** para responder ao exercício.
- Não altere os nomes dos ficheiros.
- No final, feche todas as janelas e faça **logout**, mas não desligue o PC.

1. **(20 valores)** Nas urgências dos hospitais encontra hoje um processo de triagem de doentes conforme a gravidade da sua situação clínica de acordo com uma avaliação clínica feita por um agente de saúde. Neste exercício deverá implementar a Triagem de Manchester:

O Triagem Manchester classifica os pacientes com uma de 5 cores:

- Vermelho – Emergência / Atendimento Imediato
- Laranja – Muito Urgente / Tempo espera 10 min
- Amarelo – Urgente / Tempo espera 60 min
- Verde – Pouco Urgente / Tempo espera 120 min
- Azul – Não Urgente / Tempo espera 240 min

| COR | SINTOMA |
|----------|---|
| Vermelho | Queimaduras com mais de 25% Trauma Cranio encefálico Desconforto respiratório grave Dor no peito associada a falta de ar |
| Laranja | Cefaleia intensa Dor severa Hemorragia moderada Arritmia |
| Amarelo | Convulsão Desmaios Vômitos intensos Crise de Pânico |
| Verde | Estado febril Dor leve Náusea e tontura Torcicolo |
| Azul | Aplicação de medicação Queixa crónica sem alterações |

- a. **(5 valores)** Encontra no ficheiro `pacientes.json` vários pacientes e uma lista de sintomas para cada um. Escreva uma função (de seu nome **triagem**) que leia o ficheiro e classifique segundo a Triagem de Manchester cada paciente. A função deverá retornar um dicionário com o nome da pessoa e a cor atribuída. A tabela anterior encontra-se disponível como ficheiro JSON em `cores.json`.
- b. **(5 valores)** Escreva uma função (de seu nome **ordenar**) que ordene os pacientes por prioridade de atendimento.
- c. **(5 valores)** Escreva uma função (de seu nome **tempo_espera**) que calcula quanto tempo leva a consulta de urgência de cada um dos pacientes. A função deve devolver um dicionário que associa a cada paciente o seu tempo de atendimento (representado por um

inteiro, em minutos). Considere a seguinte tabela (também disponível no ficheiro `sintomas.csv`):

| SINTOMA | TEMPO |
|--------------------------------------|---------|
| Queimaduras com mais de 25% | 50min |
| Cefaleia intensa | 35min |
| Trauma Crânio encefálico | 55min |
| Desconforto respiratório grave | 15min |
| Dor no peito associada a falta de ar | 30min |
| Dor severa | 20min |
| Hemorragia moderada | 15min |
| Arritmia | 2h00min |
| Convulsão | 3h00min |
| Desmaios | 2h30min |
| Vómitos intensos | 1h00 |
| Crise de Pânico | 30min |
| Estado febril | 15min |
| Dor leve | 5min |
| Náusea e tontura | 10min |
| Torcicolo | 5min |
| Aplicação de medicação | 5min |
| Queixa crónica sem alterações | 5min |

d. **(5 valores)** Considerando que apenas existe um médico na urgência e que todos pacientes chegaram ao mesmo tempo ($t = 0$), escreva uma função (de seu nome **stats**) que calcule algumas estatísticas:

- Quanto tempo espera cada um dos pacientes.
- Qual o tempo médio de espera por prioridade

No final o seu programa deverá produzir o seguinte output:

```
Triagem:
Teresa Gomes -> Laranja
Marta Rodrigues -> Laranja
...
Teresa Ferreira -> Laranja
Artur Fonseca -> Amarelo

Ordem de atendimento:
1º: Helena Norton
2º: Filipa Costa
...
99º: Marco Silva
100º: Helena Nunes

Tempo de espera por paciente:
Helena Norton: 0min
Filipa Costa: 50min
...
Marco Silva: 4085min
Helena Nunes: 4090min

Tempos Médios por cor:
Vermelho 383
Laranja 1299
Amarelo 2935
Verde 3952
Azul 4072
```