

ExtraX - Exercícios extra

1. Uma empresa de telecomunicações precisa de um programa para registar e faturar chamadas telefónicas feitas pelos seus clientes. Cada chamada telefónica tem uma origem e um destino, que são números de telefone, e uma duração em segundos.

- a. O programa deve apresentar um menu e processar cada opção escolhida.

```
1) Registar chamada
2) Ler ficheiro
3) Listar clientes
4) Fatura
5) Terminar
Opção?
```

- b. Faça uma função para validar um número de telefone. Um número de telefone válido é uma *string* composta de pelo menos 3 dígitos (0 a 9), que pode conter também o símbolo '+' no início. Qualquer outro carácter é proibido.

- c. Implemente a opção de registar uma nova chamada (via teclado). Use a função anterior para validar os números de origem e destino.

```
Opção? 1
Telefone origem? 972323122
Telefone destino? +12
Telefone destino? +123
Duração (s)? 100
```

- d. Implemente a opção de ler um ficheiro de chamadas. No ficheiro cada chamada é indicada por 3 “palavras” separadas por espaços em branco: a primeira é o número de origem; a segunda o número de destino; a última é a duração em segundos. Pode usar os ficheiros fornecidos para testar. Não é preciso validar os dados no ficheiro.

```
Opção? 2
Ficheiro? chamadas1.txt
```

- e. A opção Listar clientes deve apresentar a lista **ordenada** dos clientes que fizeram chamadas. Ou seja, listar todos os números de origem de chamadas, sem repetições.

```
Opção? 3
Clientes: 913862602 914293467 919274650 930930597 934863725 939999868 960373347
961393096 962613058 963970864
```

- f. Implemente a opção para gerar uma fatura detalhada. Deve pedir o número do cliente e listar todas as chamadas que fez, a sua duração e custo, bem como o custo total. O custo de cada chamada é calculado com base neste tarifário.

Destino	Custo por minuto*
Para rede fixa (números começados por “2”)	0.02 €
Internacionais (números começados por “+”)	0.80 €
Para a mesma rede (2 primeiros dígitos iguais)	0.04 €
Outros destinos**	0.10 €

* Faturado ao segundo.

** Use esta tarifa para todas as chamadas, se não as conseguir diferenciar.

Exemplo de interação esperada (dados introduzidos indicados em **bold**):

Opção? **4**

Cliente? **960373347**

Fatura do cliente 960373347

Destino	Duração	Custo
+31765214531	60	0.80
960703223	90	0.06
+81728423983	30	0.40
271055066	330	0.11
919530258	263	0.44
919530258	488	0.81
237395357	112	0.04
	Total:	2.66

2. O totobola é um jogo social em que os jogadores preenchem um boletim com uma aposta para cada jogo de futebol de uma jornada (1 – vitória da equipa anfitriã, X – empate, 2 – vitória da equipa visitante). Crie uma aplicação que permita jogar uma versão simplificada do totobola. Nesta versão, o jogador deverá preencher um boletim com um conjunto de 9 apostas correspondentes a uma única jornada do campeonato.
- a. É fornecido um ficheiro com a lista das jornadas e respetivos jogos. Cada linha indica a jornada, a equipa anfitriã e a equipa visitante de um certo jogo.

```
Jornadas.csv
1,Academica,Sp Lisbon
1,Arouca,Estoril
1,Benfica,Pacos Ferreira
1,Gil Vicente,Guimaraes
...
```

Crie uma interface com o utilizador (apostador) que permita registar as apostas para cada jogo de uma determinada jornada. O utilizador introduz o número da jornada e depois a aposta para cada jogo que o programa pede. O programa deve validar o número da jornada (que tem de ser uma das presentes no ficheiro `Jornadas.txt`) e cada aposta (só permitindo 1, X ou 2). A interação deve seguir o exemplo abaixo (input indicado em **bold**). No final deverá guardar um ficheiro com o nome igual ao número da jornada (exemplo: `jornada1.csv`) com as várias apostas feitas uma por linha. Cada linha deverá ter o número do jogo e a aposta (1, X ou 2).

```
Interface:
Jornada? 14
Jornada inválida!
Jornada? 13
1 Belenenses vs Sp Braga: 1
2 Gil Vicente vs Academica: 3
Aposta inválida!
2 Gil Vicente vs Academica: 2
3 Guimaraes vs Rio Ave: X
...
jornada13.csv:
1, 1
2, 2
3, X
...
```

- b. Também é fornecido um ficheiro com os jogos e resultados da 1ª Liga portuguesa da época 2014-15 realizados até ao momento. Cada linha contém a informação de um jogo: data, equipa anfitriã, equipa visitante, golos da anfitriã, golos da visitante.

```
Jogos.csv
15/08/14,Porto,Maritimo,2,0
16/08/14,Academica,Sp Lisbon,1,1
16/08/14,Gil Vicente,Guimaraes,1,3
...
```

Altere agora o programa para que imediatamente após o jogador ter preenchido o seu boletim, apresente uma tabela com os jogos dessa

jornada, seus resultados, apostas feitas, e indicar quais acertou/errou. Tome atenção à formatação, que deve seguir o exemplo abaixo.

```
Jornada 13
1          Setubal  0-1  Boavista          : 1  (ERRADO)
2          Belenenses 0-1  Sp Braga          : 2  (CERTO)
3          Gil Vicente 1-1  Academica          : X  (CERTO)
4          Pacos Ferreira 2-1  Arouca          : X  (ERRADO)
5          Penafiel 2-1  Nacional          : X  (ERRADO)
6          Guimaraes 0-0  Rio Ave          : X  (CERTO)
7          Maritimo 0-0  Estoril          : X  (CERTO)
8          Porto 0-2  Benfica          : X  (ERRADO)
9          Sp Lisbon 1-1  Moreirense          : X  (CERTO)
```

- c. No fim da tabela, indique o número de apostas certas e se conseguiu o 1º prémio (todos certos), o 2º prémio (8 certos), o 3º prémio (7 certos), ou se não tem prémio.

```
...
TEM 5 CERTAS. SEM PRÉMIO.
```

- d. Repita o jogo até que utilizador responda 0 (zero) à questão “Jornada?”. Sabendo que cada boletim custa 0.40 euros, que o 1º prémio vale 5000€, o 2º prémio vale 1000€ e o 3º prémio vale 100€, calcule o saldo do jogador no final de cada boletim. O saldo inicial do jogador é 0, pelo que se preencher um boletim fica imediatamente com um saldo negativo de -0.40€.

```
...
Saldo: -0.40 euro
```

- e. Acrescente a possibilidade de fazer apostas múltiplas. Neste caso, o utilizador pode introduzir uma dupla (1X, X2, 12) ou tripla (1X2) em cada jogo. Após ler um boletim com apostas múltiplas, calcule o número equivalente de apostas e o seu custo (cada boletim custa 0.40 euros). Note que por cada aposta dupla, o número equivalente de apostas duplica, por cada aposta tripla, triplica. Por exemplo:

Apostas Simples	Apostas Duplas	Apostas Triplas	Nº equivalente de apostas	Custo (€)
9	0	0	$1^9 = 1$	0.40
8	1	0	$1^8 \times 2^1 = 2$	0.80
8	0	1	$1^8 \times 2^0 \times 3^1 = 3$	1.20
7	2	0	$1^7 \times 2^2 \times 3^0 = 4$	1.60
7	1	1	$1^7 \times 2^1 \times 3^1 = 6$	2.40
6	3	0	$1^6 \times 2^3 \times 3^0 = 8$	3.20
7	0	2	$1^7 \times 2^0 \times 3^2 = 9$	3.60
6	2	1	$1^6 \times 2^2 \times 3^1 = 12$	4.80

3. Vamos recriar a caixa registadora de um hipermercado. A caixa registadora tem uma base de dados (ficheiro de texto) com o código do produto, nome do produto, secção a que pertence, o preço bruto e taxa de IVA. Por exemplo:

BaseDados.txt:

```
311;Papel Higiénico; Merceria; 3.99; 23%
814;Carne Picada; Talho; 2.99; 5%
351;Arroz; Merceria; 1.00; 5%
562;Pão de Forma; Padaria; 1.99; 5%
```

O programa deverá apresentar repetidamente um menu com as seguintes opções:

```
(I)nserir itens
(F)acturar
(S)air
Opção?
```

- a. Na opção de “Inserir Itens” deverá pedir o código do produto e calcular o seu preço final imprimindo no ecrã algo como:

```
Arroz: 1.23€
```

A inserção deverá ser repetida até que o utilizador digite o código especial 0 (zero). Itens que não correspondam a um item na base de dados devem ser ignorados.

- b. Quando o utilizador optar por sair, deverá registar num ficheiro todas as vendas feitas pela caixa. Cada linha corresponderá a um cliente e registará a data da compra (use `time.strftime("%d/%m/%Y %H:%M")`) e o valor final da venda. Exemplo:

Vendas.txt:

```
2014-12-10 15:50: 120.12
2014-12-10 16:03: 73.99
2014-12-10 16:07: 33.00
```

- c. Quando o utilizador optar por sair, deverá registar noutro ficheiro o stock vendido. Cada linha terá um código de um produto e o número total de unidades vendidas desse produto). Exemplo:

StockOut.txt:

```
311; 0
814; 1
562; 3
```

- d. Quando o utilizador pedir factura, deverá imprimir uma factura de todos produtos organizados por secção e ordenados por nome. Exemplo:

```
Mercearia:
  2 Arroz                (IVA 5%)    1,23€
  1 Papel Higiénico      (IVA 23%)   4,90€
Talho:
  1 Carne Picada         (IVA 23%)   3,14€
Total Bruto:             7,97€
Total IVA:                1,30€
Total Liquido:           9,27€
```

4. Pretende-se fazer um histograma das 10 palavras mais mencionadas no twitter, num dado período de tempo. Para tal, é-lhe fornecido um ficheiro `twitter.json` contendo uma lista de 895 twitts em formato JSON. O programa `twitter.py` já contém instruções para ler esse ficheiro e mostrar o conteúdo de alguns dos dados lidos. Analise-o e altere-o para resolver o exercício.
- Crie uma lista de todas as palavras mencionadas ('text') nos twits.
 - Ordene essa lista por ordem crescente de número de vezes que a palavra é dita (não poderá recorrer aos métodos de ordenação do Python Terá que implementar o seu próprio método).
 - Crie uma nova lista ordenada apenas com as hashtags (palavras começadas por #).
 - Crie um histograma normalizado a 18 colunas das hashtags mais populares (significa que a hashtag mais popular tem tamanho 18 e as seguintes são proporcionais a esse tamanho). Exemplo:

```
#PeC (100) ++++++
#SS5 (61)  ++++++
#FX (44)   ++++++
#TVK (43)  ++++++
```