

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA</p> <p><i>Departamento de Engenharia Informática</i></p>	<p align="center">Trabalho nº 1B/C de Algoritmos e Estruturas de Dados 2017-2018 – 2º Semestre</p> <p>Data-limite de submissão no Mooshak:</p> <ul style="list-style-type: none"> • Tarefa A, B, C: 17 de Março de 2018, 23h55. • Tarefa D, E: 26 de março de 2018, 23h55. <p>Data-limite de entrega do relatório:</p> <ul style="list-style-type: none"> • 03 de Abril de 2018, 9h30. <p>Em papel, manuscrito e entregue no cacifo do respetivo docente das aulas TP.</p>
<p><i>Nota Importante: A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude pode levar a anulação da componente prática tanto do facilitador como do prevaricador.</i></p>	

Objetivos: Árvores de Pesquisa.

Notas Importantes:

- O relatório do trabalho deve ser realizado, MANUSCRITO, no formulário que será oportunamente fornecido.
- A não entrega do relatório do trabalho no prazo-limite estabelecido implica que o aluno tem uma classificação de ZERO na avaliação deste trabalho.

As várias tarefas incluídas neste Trabalho Prático correspondem à resolução de um mesmo problema (descrito nas páginas seguintes) usando diferentes alternativas de implementação:

- Tarefa A – Solução baseada em árvores binárias de pesquisa (usando e/ou adaptando o código-fonte já fornecido pelos docentes)
- Tarefa B – Solução baseada em árvores AVL
- Tarefa C – Solução baseada em árvores Vermelhas e Pretas
- Tarefa D – Solução baseada numa das seguintes soluções:
 - Árvores Aleatórias ou Listas de Saltos
- Tarefa E – Solução baseada em *Splay Trees*

Descrição do Problema – “Passageiro Frequente”

O grupo *Star Alliance* assumiu recentemente a responsabilidade de gerir de forma conjunta os programas de passageiro frequente das diversas companhias aéreas que a integram, de modo a que os passageiros frequentes de uma das companhias possam facilmente acumular e gastar pontos quando viajam com as outras companhias do grupo.

Foi-lhe pedido para elaborar uma aplicação que mantenha um registo de acesso rápido ao valor de milhas acumuladas de cada cartão de passageiro frequente. O principal objectivo é otimizar a pesquisa do valor de milhas acumuladas por um dado cartão, dado que esta informação é pesquisada com elevada frequência no *website* da *Star Alliance* e das companhias associadas.

Cada cartão é identificado por uma *string* com menos de 15 caracteres (tal como, por exemplo “TP298421777” ou “LH2009364744”). O valor das milhas acumuladas é sempre um inteiro.

Os formatos usados para estes campos são os seguintes:

Cartão: string sem espaços vazios e com um máximo de com 15 caracteres

Milhas: valor inteiro positivo ou negativo (creditar/descontar milhas)

Input:

O seu programa deve ler uma sequência de linhas (até encontrar uma linha em branco), cada uma das quais correspondente a uma das seguintes operações:

UPDATE <Cartão> <Milhas>

Se o cartão ainda não está na base de dados é adicionado com crédito de *Milhas*.

Se o cartão já existe o montante total de milhas é atualizado com o crédito ou débito de *Milhas* (consoante o valor de *milhas* seja positivo ou negativo).

REMOVE <Cartão>

Retira o cartão da base de dados, por exemplo porque já expirou a sua validade.

Caso o cartão não conste da base de dados esta operação é simplesmente ignorada.

SALDO <Cartão>

Esta operação procura o cartão na base de dados.

Se o cartão estiver na base de dados é mostrado o saldo disponível:

<Cartão> **SALDO** <Saldo>

Se o cartão não estiver na base de dados é mostrada a seguinte informação:

<Cartão> **INEXISTENTE**

IMPRIME

Para imprimir a informação de todos os cartões que constam na base de dados, ordenados de forma crescente (por ordem alfabética) e com o seguinte formato para cada contribuinte:

<Cartão> **SALDO** <Saldo>

TERMINA

Para terminar a execução da aplicação.

Exemplo de *Input*:

```
UPDATE TP199529030 35[\n]
UPDATE TP111111111 40[\n]
SALDO TP111111111[\n]
SALDO LH222222222[\n]
UPDATE TP111111111 5[\n]
UPDATE TP456321098 10[\n]
SALDO TP111111111[\n]
UPDATE LH33333333 40[\n]
UPDATE LH33333333 -23[\n]
SALDO LH33333333[\n]
UPDATE TP111111111 10[\n]
UPDATE TP111111111 100[\n]
SALDO TP111111111[\n]
REMOVE TP111111111[\n]
UPDATE TP235888945 1[\n]
UPDATE TP456321098 10[\n]
IMPRIME[\n]
TERMINA[\n]
```

Output correspondente:

```
TP111111111 SALDO 40[\n]
LH222222222 INEXISTENTE[\n]
TP111111111 SALDO 45[\n]
LH33333333 SALDO 17[\n]
TP111111111 SALDO 155[\n]
LH33333333 SALDO 17[\n]
TP199529030 SALDO 35[\n]
TP235888945 SALDO 1[\n]
TP456321098 SALDO 20[\n]
```

Relatório

O *template* para o relatório, assim como os ficheiros a usar nas correspondentes medições de desempenho, serão posteriormente divulgados.