

Trabalho prático 3

1) Informação geral

O trabalho prático 3 consiste na implementação de programa em C conforme descrito de seguida.

Este trabalho deverá ser feito de forma autónoma por cada grupo na aula prática 12 e completado fora das aulas até à data limite estabelecida. A consulta de informação nas diversas fontes disponíveis é aceitável. No entanto, o código submetido deverá ser apenas da autoria dos elementos do grupo e quaisquer cópias detetadas serão devidamente penalizadas. A incapacidade de explicar o código submetido por parte de algum elemento do grupo implicará também numa penalização.

O prazo de submissão no Moodle de Programação 2 é 22 de Maio às 12:00 (meio-dia).

2) Descrição

Pretendes concorrer para trabalhar numa equipa de um centro financeiro que tem a tarefa de analisar todas as transações diárias e reportar em tempo real informação sobre as transações ocorridas até ao momento. Sobre cada transação, a equipa tem informação do número de ações transacionadas e a empresa em causa.

Para avaliar a tua competência para integrar a equipa, o responsável atribuiu-te duas tarefas. O Responsável da Equipa vai avaliar o teu trabalho com um programa semelhante ao **testModules.c** e vai contratar os que apresentarem melhor score final (obtido a partir do número de transações processadas em 2 segundos). A única restrição é a utilização das funções descritas em cada tarefa. Ou seja, podem ser usadas quaisquer estruturas de dados, algoritmos e bibliotecas que sejam considerados mais pertinentes.

TAREFA1: escrever um módulo que recebe o número de ações transacionadas e devolve a mediana dos valores transacionados até ao momento. Deves implementar as seguintes funções em C:

```
void median_initModule(int maxTransactions)
```

Serve para realizar as inicializações necessárias e é invocada no início da sessão. O parâmetro maxTransactions é o número máximo de transações que poderão ser processadas na sessão.

```
int median_newObservation(int n, float *numActions)
```

Invocada sempre que ocorre uma nova transação. O parâmetro numActions é o número de ações transacionadas na nova transação e o parâmetro updatedMedian devolve a mediana atualizada. A função retorna -1 em caso de erro e 1 em caso de funcionamento normal.

```
void median_closeModule()
```

Invocada no fim da sessão e serve para libertar toda a memória alocada previamente.

Por exemplo, a sequência de valores transacionados

75, 35, 51, 75, 45

deverá resultar na sequência de medianas

75.0, 55.0, 51.0, 63.0, 51.0

TAREFA2: escrever um módulo que recebe o nome da empresa transacionada e devolve o nome da empresa mais vezes transacionada (isto é, a moda) até ao momento. Podes assumir que o nome de cada empresa não tem mais do que 15 caracteres mas não sabes quantas empresas são transacionadas. Deves implementar as seguintes funções em C:

`void mode_initModule(int maxTransactions)`

Serve para realizar as inicializações necessárias e é invocada no início da sessão. O parâmetro `maxTransactions` é o número máximo de transações que terão de ser processadas na sessão.

`int mode_newObservation(const char *companyName, char *updatedMode)`

Invocada sempre que ocorre uma nova transação. O parâmetro `companyName` é o nome da empresa cujas ações foram transacionadas na nova transação e o parâmetro `updatedMode` recebe a moda atualizada (nome da empresa mais vezes transacionada até ao momento). Se houver mais do que uma moda a função deve retornar o nome da primeira empresa a atingir esse valor. A função retorna -1 em caso de erro e 1 em caso de funcionamento normal.

`void mode_closeModule()`

Invocada no fim da sessão e serve para libertar toda a memória alocada previamente.

Por exemplo, a sequência de empresas
Google, Microsoft, Microsoft, Google, Amazon
deverá resultar na sequência de modas:
Google, Google, Microsoft, Microsoft, Microsoft

3) Avaliação

A classificação do trabalho é dada pela avaliação feita à implementação submetida pelos estudantes mas também pelo desempenho dos estudantes na aula dedicada a este trabalho. A classificação final do trabalho (T3) é dada por:

$$T3 = 0.7 \text{ Implementação} + 0.1 \text{ Memória} + 0.2 \text{ Desempenho}$$

A classificação da implementação é essencialmente determinada por testes automáticos adicionais. No caso da implementação submetida não compilar, esta componente será de 0%. Haverá uma diferenciação da classificação, de acordo com o score obtido.

A gestão de memória também será avaliada, sendo considerados 3 patamares: 100% nenhum *memory leak*, 50% alguns *memory leaks* mas pouco significativos, 0% muitos *memory leaks*.

O desempenho será avaliado durante a aula e está dependente da entrega do formulário "Preparação do trabalho" que se encontra disponível no Moodle. A classificação de desempenho poderá ser diferente para cada elemento do grupo.

4) Submissão da resolução

A submissão é apenas possível através do Moodle e até à data indicada no início do documento. Deverá ser submetido um ficheiro *zip* contendo:

- ficheiro **modules.c**
- **outros** ficheiros **.c/.h** que tenham sido utilizados como bibliotecas adicionais
- ficheiro **autores.txt**, indicando o nome e número dos elementos do grupo

Nota importante: apenas as submissões com o seguinte nome serão aceites:
`T3_G<numero_do_grupo>.zip`. Por exemplo, `T3_G999.zip`