

Métodos de Programação 2

MiETI :: 1ºano

Trabalho Prático 1: Arrays, estruturas e ficheiros

Ano lectivo 2017/18

1 Objectivos e Organização

Este trabalho prático tem como principais **objectivos**:

- aumentar as competências em programação imperativa;
- familiarização com ferramentas de apoio à programação, vulgarmente conhecidas como IDEs. Dois exemplos são o CodeBlocks ou o Visual Studio Code;
- aumentar a capacidade de trabalhar com *arrays* para armazenar coleções (conjuntos ou sequências) de elementos do mesmo tipo;
- praticar a utilização de estruturas para armazenar informação relativa a uma dada entidade, usando tipos distintos (para construir fichas ou registos);
- praticar o uso de ficheiros para guardar de forma permanente as coleções contidas na memória central, normalmente em arrays ou estruturas.

O trabalho será realizado em grupos de 2 alunos.

Este documento contém 4 enunciados, dos quais cada grupo deverá resolver **um**, escolhido ao gosto do grupo.

Neste primeiro TP, que se pretende que seja resolvido em aproximadamente 2 semanas, os resultados pedidos são simples: aprecia-se a criatividade dos grupos ao incluir outras funcionalidades (resultados) além dos pedidos.

Cada grupo deve entregar, através de um link a anunciar brevemente, a sua solução **até Domingo dia 17 de Março**. **NÃO** submetam um ficheiro comprimido (ZIP, RAR, etc), nem enviem vários ficheiros. Submetam apenas o ficheiro PDF do relatório, que deve conter toda a informação sobre o trabalho realizado. O ficheiro PDF a submeter deve ter o nome `mp21718-TP1-grupoNN.pdf`, em que NN é o número do Grupo.

O programa desenvolvido será apresentado aos membros da equipa docente, totalmente pronto e a funcionar (acompanhado do respectivo relatório de desenvolvimento) e será defendido por todos os elementos do grupo, em data a marcar.

O **relatório** a elaborar (em WORD ou L^AT_EX) deve ser claro e incluir o enunciado, a descrição do problema, as decisões que lideraram o desenho da solução e da sua implementação, o código C da implementação e exemplos de utilização da solução desenvolvida (dados de entrada e respectivo resultado produzido).

2 Enunciados

Antes de fazer a sua escolha, leia atentamente cada um dos enunciados abaixo e escolha aquele que lhe oferecer maior desafio.

Não hesite em falar com qualquer um dos dois docentes da UC para perceber bem cada enunciado – parta para a resolução com todas as dúvidas completamente esclarecidas para que na apresentação não surjam surpresas.

2.1 Classificação de Atletas numa Competição de Ginástica

Neste problema pretende-se ordenar os atletas de uma competição desportiva por ordem decrescente da pontuação final obtida em 8 provas realizadas (cada prova avaliada com uma nota de 0 a 10), sendo que a pontuação final é a média aritmética das 8 pontuações que ficam depois de retirar do conjunto a maior nota e a menor nota.

O seu programa deve ler os nomes de cada atleta e a seguir a cada nome as 10 pontuações obtidas em cada prova. Compete ao grupo decidir quando termina a introdução dos dados que devem ficar internamente armazenados em `'arrays'`.

Os dados introduzidos devem ser gravados num ficheiro binário para que possam ser recuperados numa sessão de trabalho posterior.

Os resultados produzidos devem ser enviados para o `standard output` com o nome e a classificação devidamente formatados.

2.2 Gestão das Temperaturas Mensais

Neste problema pretende-se ler as temperaturas máxima e mínima (a variar de -10 graus a +40 graus) de cada dia de um mês, devendo esta operação ser realizada para vários meses de um ano (que tem de ser registado) de modo a poder-se fazer um gráfico de barras com a temperatura média de cada dia para um mês escolhido pelo utilizador. Adicionalmente também se pretende calcular a temperatura média das máximas e mínimas de um dado dia de todos os meses registados.

O seu programa deve ler o nome do mês e a seguir temperaturas máxima e mínima de cada um dos seus dias. Compete ao grupo decidir quando termina a introdução dos dados que devem ficar internamente armazenados em `'arrays'`.

Os dados introduzidos devem ser gravados num ficheiro binário para que possam ser recuperados numa sessão de trabalho posterior.

Os resultados produzidos devem ser enviados para o `standard output` devidamente formatados quando for caso disso. Para facilitar a programação, desenhe o gráfico deitado e use os caracteres `'+'` ou `'*'` para fazer o preenchimento das barras de acordo com uma escala escolhida.

2.3 Escolha de Artigos submetidos a uma Conferencia

Neste problema pretende-se criar um sistema que decida quais os artigos a aceitar para uma conferência dadas as pontuações atribuídas a cada artigo por cada um dos 4 Revisores (que leram o artigo e procederam à sua pontuação). A classificação final do artigo é dada pela média pesada das pontuações dos revisores, usando a segurança do Revisor (um valor no intervalo 1 a 3) como peso a afetar cada nota. Após calcular a classificação final, os artigos são ordenados por ordem decrescente e serão aceites todos os que tenham nota superior ou igual a um valor mínimo definido pelo gestor da conferência.

O seu programa deve ler o identificador do artigo (uma simples sigla) seguido das 4 classificações obtidas (um número inteiro de -3 a 3). Compete ao grupo decidir quando termina a introdução dos dados que devem ficar internamente armazenados em `'arrays'`.

Os dados introduzidos devem ser gravados num ficheiro binário para que possam ser recuperados numa sessão de trabalho posterior.

Os resultados produzidos (lista dos aceites) devem ser enviados para o `standard output` mostrando a sigla do artigo e sua classificação final devidamente formatados.

2.4 Gestão de Salas num Complexo Pedagógico

Neste problema pretende-se ajudar o gestor de um complex pedagógico, responsável por fazer horários de uma escola, armazenando para cada sala do complexo a sua ocupação de hora a hora entre as 8h e as 20h em cada dia de 2^{af} a 6^{af}. Por cada hora deverá ser registado se a sala está L(ivre), ou O(cupada) e nesse caso deve conter a sigla da UC marcada para essa hora e a sigla do curso. O objetivo é que o sistema responda ao utilizador qual o estado de uma dada sala, numa dada hora e num dado dia (indicados pelo utilizador), ou que liste todas as salas livres a determinada hora de um dado dia (indicados pelo utilizador).

O seu programa deve ler o identificador da sala (uma simples sigla) e por cada dia útil da semana, lerá a hora e a UC/Curso por cada marcação desse dia. Compete ao grupo decidir quando termina a introdução dos dados que devem ficar internamente armazenados em `'arrays'`.

Os dados introduzidos devem ser gravados num ficheiro binário para que possam ser recuperados numa sessão de trabalho posterior.

Os resultados produzidos (estado duma sala ou lista de salas livre) devem ser enviados para o `standard output` devidamente formatados.