

	Departamento de Matemática da Universidade de Coimbra		
Duração: 2h30m	Métodos de Programação II	Exame de Recurso AP	08/02/2007

1) Neste exercício, vamos trabalhar com elementos inteiros e não-negativos, $k \in \mathbb{N}_0$.

a) Escreva **declarações** em Pascal para definir completamente uma tabela bidimensional (matriz) de elementos inteiros e não-negativos, a que chamará **Matriz**, e as respectivas dimensões máximas, as quais deverão ser denominadas **MaxLin** e **MaxCol**.

Note que deve usar **declarações** que permitam **alterações** posteriores fáceis, caso seja necessário aumentar (ou diminuir) a quantidade de elementos na tabela.

b) Elabore um procedimento para, dada uma tabela do tipo **Matriz**, **MX**, inicializar todo o elemento da matriz com o **valor** zero.

c) Elabore um procedimento para, a partir do teclado *input standard*, ler uma tabela do tipo **Matriz**, **ML**, sem perguntar quantos elementos vão ser lidos (pode ler cada linha do teclado como uma linha da matriz e usar uma sentinela para sinalizar o fim da leitura).

Note que:

- deve usar a alínea anterior
- deve devolver a tabela **ML** bem **como** as respectivas dimensões úteis (ou seja, o maior índice de linha e o maior índice de coluna).

Por exemplo, se a matriz lida tivesse o seguinte aspecto:

$$ML = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 0 & 0 & \dots & 0 \\ 2 & 1 & 0 & 11 & 23 & 0 & 0 & \dots & 0 \\ 0 & 22 & 31 & 4 & 30 & 55 & 0 & \dots & 0 \\ 0 & 0 & 122 & 34 & 5 & 0 & 0 & \dots & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

a dimensão útil para as linhas seria 5 e para as **colunas** seria 6.

d) Construa um sub-programa para, dada uma tabela do tipo **Matriz**, **MS** e as **respectivas** dimensões úteis, devolver o valor da soma de todos os elementos úteis da matriz.

2) Considere as seguintes **declarações** de tipos em Pascal para implementar uma lista de alunos:

```

type
  Notas = array real;
  Aluno = record
    nome: packed array[1..60] of char;
    tipoaval : boolean;
    avalia: array[1..31] of Notas;
  end;
  Lista = array [1..100] of Aluno;

```

Onde **tipoaval** é TRUE se o aluno for de **avaliação** contínua e **FALSE** caso contrário.

- a) Explique **quais** as **vantagens/desvantagens** entre implementar uma lista como uma tabela (estrutura estática) ou como uma lista ligada (estrutura dinâmica).
- b) Modifique, no que for necessário, as **declarações** apresentadas, de modo a alterar o tipo Lista de tabela para lista ligada simples de Aluno.

Nas alíneas restantes, considere sempre o uso de uma representação de lista ligada.

- c) Supondo que, no caso de avaliação contínua:
 - **avalia[1]** tem uma nota de avaliação contínua
 - **avalia[2]** tem uma nota de exame em época normal
 - **avalia[3]** tem uma nota de exame em época de recurso

e que a nota final deve ser calculada do modo seguinte: se a nota de avaliação contínua é superior ou igual a 2,5 e se a nota de exame está entre 7 e 15, então, a nota final é a soma arredondada das duas notas. Caso contrário, o aluno reprova.

Considerando também que o aluno **que não** fez avaliação contínua é avaliado do modo habitual, elabore um módulo para efectuar a listagem das notas finais de todos os alunos. Por exemplo:

Nome	Nota Final	Aval. Cont.
Ana Beatriz Bobadela	18	S
Antonino Virtual	R	N
Paulo Regio da Silva	15	N
Renardo Limiano	17	S
...		

Note que se supõe que já foram realizadas as duas épocas de exames e deve ser **listada** a melhor das duas notas (considera-se que, se o aluno faltou ou desistiu, estará o valor 0 como nota).

- d) Escreva um subprograma para, dado **nomX**, um nome de aluno e uma Lista, procurar e remover o elemento dessa lista cujo nome é igual ao dado. Caso **não** seja encontrado este nome, a lista deve ficar inalterada.

3)

- a) Dê um exemplo de uma Estrutura de Dados do tipo Fila, dizendo qual a propriedade que define essa estrutura e qual o conjunto de **operações** mínimo para definir a estrutura
- b) Qual acha ser a **representação** em Pascal (com que tipos de dados) mais adequada para implementar uma estrutura de Fila e porquê.
- c) Usando dados do tipo ponteiro, declare uma representação para implementar uma estrutura de dados do tipo Fila.
- d) Usando a **alínea** anterior, implemente a operação que permite **colocar um novo** elemento numa Fila.
- e) Implemente agora a operação que permite listar por ordem inversa **em l 9** uma dada Fila, F.

4)

- a) Explique quando e como se pode usar a **instrução** go to num programa. (Tente dar pequenos exemplos ilustrativos)
- b) Explique o que se entende por estratégia de resolução "Falha e Erro" (**Trial-and-error**), ilustrando com um exemplo de entre aqueles que foram apresentados nas aulas.

