

	Departamento de Matemática da Universidade de Coimbra		
Duração: 2h30m	Métodos de Programação II	Exame AP	26/01/2007

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 unidimensional (vector) de elementos inteiros e não-negativos, a que chamará `Vector`, e a respectiva dimensão máxima, a que deverá chamar `MaxVec`.

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

- b) Considere dada uma tabela do tipo `Vector`, `vec`, com  $n$  elementos inteiros e não-negativos, guardados a partir do índice inicial da tabela. Implemente um algoritmo que **devolva** os valores do **maior e do menor** de todos os elementos do vector `vec` dado.

- c) Elabore um procedimento para, dado um número **inteiro e não-negativo** através do teclado input *standard*, estabelecer a **representação** decimal desse número num vector do tipo `Vector`, chamado `num`, onde, obviamente, todos os elementos de `num` serão um dos possíveis dez algarismos:  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .

Note que deve devolver o vector `num` com a **representação** do número e **ainda** a respectiva dimensão útil.

- d) Construa um procedimento para, dado um número natural não-negativo num vector como em (c), e a respectiva dimensão, e dado um outro algarismo qualquer,  $p \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ , devolver, caso seja possível, o vector resultante do produto de  $p$  por `d`, mantendo a representação decimal tal como em c).

- e) Imagine agora que precisa de passar a inteiros com sinal. Relembrando que, como vimos nas alíneas anteriores, a dimensão do vector é parte integrante da representação interna desse número, proponha novas declarações em Pascal que permitam implementar **fielmente** esta nova representação.

2) Considere as seguintes declarações de tipos em Pascal:

```

const Limite = 100;
type
  NTipos = (NReal, NComplexo);
  Complexo = record
    re, im: real;
  end;
  Numeros = record case tipo: NTipos of
    NReal: (re: real);
    NComplexos: (z: complexo);
  end;

```




Lista = array[1..Limite] of Numeros;

- a) Escreva um procedimento para, dados dois "números" do tipo **Numeros**, efectuar a sua soma, devolvendo o *resultado num registo do mesmo tipo*.
- b) Modifique, no que for necessário, as declarações **apresentadas**, de modo a alterar o tipo Lista de tabela de **Numeros** para lista **ligada** simples de **Numeros**.
- c) Utilizando o módulo construído na alínea (a) e a nova declaração de Lista, elabore um novo módulo para efectuar a soma de todos os elementos de uma Lista dada, e devolver o valor obtido.
- d) Escreva um subprograma para, dado k e uma **Lista**, procurar e remover o elemento que está na k-ésima posição dessa lista. Caso a lista tenha menos que k elementos, deve ficar **inalterada**.

3)

- a) Diga em que consiste uma *Estrutura de Dados*?
- b) Dê um exemplo de uma Estrutura de Dados do tipo *Pilha*, dizendo qual a propriedade que define essa estrutura, qual o conjunto de operações mínimo para definir a estrutura e como acha que poderia **representá-la** em Pascal (com que tipos de dados).
- c) Qual a principal diferença entre uma Estrutura de Dados do tipo *Pilha* e uma estrutura do tipo *Fila*?
- d) Usando dados do tipo ponteiro, declare uma representação para implementar uma estrutura de dados do tipo *Fila*.
- e) Usando a alínea anterior, **implemente** a operação que permite colocar um novo elemento numa *Fila*.

4)

- a) Explique, succintamente, qual o significado de dizer que "o algoritmo X é **mais** eficiente que o algoritmo Y" para resolver um dado problema.
- b) Explique o que se entende por estratégia de resolução "dividir-e-conquistar", ilustrando com um exemplo **simples** de entre os apresentados nas aulas.

D D D