Tipos Abstractos de Dados(2008/2009)

Exercícios sobre Listas Ligadas Lineares

1. Considere o tipo Lista Ligada Linear, definido por:

```
type ponteiro = ^elemento;
     elemento = record num : integer;
                       prox : ponteiro
                end;
```

e um Ficheiro `numeros.txt` que contém números inteiros, um por cada linha. Construa os procedimentos:

(a) `Registar`: para registar no Ficheiro os números contido numa Lista, pela mesma ordem.

(b) `RegistarR`: para registar no Ficheiro os números contido numa Lista, por ordem inversa.

(c) `RecuperarR`: para construir uma Lista com os números registados no Ficheiro, por ordem inversa.

(d) `Recuperar`: para construir uma Lista com os números registados no Ficheiro, pela mesma ordem.

(e) `DarMax`: que devolve os ponteiros `pmax` para o nó que contém o elemento máximo da Lista e `pantmax` para o nó anterior.

(f) `RemoverMax`: remove e destrói o nó com o elemento máximo da Lista, numa só passagem e sem utilizar o procedimento anterior.

(g) `RemoverDarMax`: remove e devolve o ponteiro `pmax` para o nó com o elemento máximo da Lista.

(h) `OrdenarLista`: utilizando o procedimento anterior ordenar a Lista dada, sem criar espaço de memória adicional.

2. São bem conhecidas as limitações do tipo `integer` quanto à grandeza dos seus valores. Para permitir a manipulação de números inteiros de *grandeza astronómica*, podemos utilizar uma representação na forma de Lista Ligada Linear,

```
type tipodigito = 0..9;
     astronomico = ^elemento;
     elemento = record digito : tipodigito;
                       prox : astronomico
                end;
```

onde o primeiro elemento da lista contém o algarismo das unidades do inteiro representado.

Comecemos por construir os módulos:

(a) `function representacao(n : integer) : astronomico;`

(b) `procedure LereConstruir(var a : astronomico);`

(c) `procedure Escrever(a : astronomico);`

(d) `function soma(a, b : astronomico) : astronomico;`

(e) `function diferenca(a, b : astronomico) : astronomico;`

(f) `function iguais(a, b : astronomico) : boolean;`

(g) `function maior(a, b : astronomico) : boolean;`

(h) `function copia(a : astronomico) : astronomico;`

(i) `function zero : astronomico;`

(j) ...

3. Pelo Teorema Fundamental da Aritmética qualquer número inteiro positivo pode ser decomposto como produto de potências não negativas de números primos, $n = p_1^{n_1} \times p_2^{n_2} \ldots \times p_k^{n_k}$.

   Considere os tipo de dados em Pascal:

```
type canonico =^termo;
     termo= record
                primo, expoente:integer;
                prox: canonico
     end;
```

   Comece por fazer um esboço da representação do número 3528 segundo a definição anterior.

   Elabore subprogramas para:

   (a) obter a representação canónica de um dado inteiro positivo x.

   (b) verificar se um dado inteiro na forma canónica é ou não um número primo.

   (c) calcular o produto de dois inteiros na forma canónica.

   (d) calcular o máximo divisor comum de dois números inteiros na forma canónica.

   (e) calcular o mínimo múltiplo comum de dois números inteiros na forma canónica.