

# Computação e Programação

(LEA, LEB, LEGI, LEMa, LEGM, LEQ, LQ)

## Exame 2A

Departamento de Matemática, IST

11 de Fevereiro de 2004, 9h00m

Duração: 2h30m

### Grupo I

[1,5 + 1,5]

a) Desenvolva em MATLAB uma função  $f$  que recebe como argumento um vector de números inteiros, percorre o vector dado recorrendo a um ciclo `while` e determina o par  $[x, y]$  em que  $x$  é o maior número par e  $y$  é o menor número ímpar do vector.

b) Desenvolva em MATLAB uma função  $g$  que recebe como argumento uma matriz de números inteiros, percorre a matriz dada elemento a elemento recorrendo a dois ciclos `while` encaixados e determina o par  $[x, y]$  em que  $x$  é o maior número par e  $y$  é o menor número ímpar da matriz.

### Grupo II

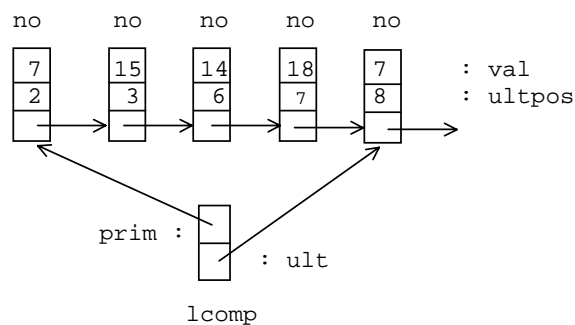
[2,0 + 2,0]

a) Por *lista com representação compactada* (`lcomp`) entende-se uma lista em que a ocorrência de elementos consecutivos iguais é representada de uma forma compacta, indicando apenas o elemento e a última posição em que ele ocorre na lista. Por exemplo, a lista  $\{7, 7, 15, 14, 14, 14, 18, 7\}$  é representada de forma compacta por  $\{(7, 2), (15, 3), (14, 6), (18, 7), (7, 8)\}$ , ou seja, o número 7 ocorre até à segunda posição, em seguida o número 15 ocorre até à terceira, em seguida o número 14 ocorre até à sexta e assim sucessivamente. Escolhendo a implementação seguinte para o tipo `lcomp`

```
type, public :: lcomp
  private
    type(no), pointer :: prim
    type(no), pointer :: ult
end type lcomp
```

```
type, private :: no
  integer :: val
  integer :: ultpos
  type(no), pointer :: seg
end type no
```

a lista anterior é implementada por:



Desenvolva em F um módulo que disponibilize as seguintes funções e subrotinas sobre `lcomp`:

- `nova()`: função que retorna a lista sem elementos;
- `app(x,w)`: subrotina que insere o elemento `x` no fim da lista `w`. Note que ao inserir um elemento na lista, se este for igual ao último elemento, apenas há que incrementar a última posição em que ele ocorre;
- `rest(w)`: subrotina que apaga o primeiro elemento da lista `w`, desde que `w` não seja a lista vazia. Note que ao apagar o primeiro elemento, se ele aparecer repetido mais do que uma vez, apenas há que decrementar a última posição em que ele ocorre. Em qualquer dos casos, há que decrementar uma unidade nas posições dos restantes nós.
- `prim(w)`: função que devolve o primeiro elemento da lista `w`, desde que `w` não seja a lista vazia;
- `vaziaQ(w)`: função que devolve `.true.` se `w` for a lista vazia e `.false.` caso contrário;
- `comp(w,n)`: subrotina que devolve no parâmetro de saída `n` o comprimento da lista `w`;
- `val(w,i,x)`: subrotina que devolve no parâmetro de saída `x` o valor do `i`-ésimo elemento da lista `w`, desde que `w` seja uma lista não vazia e `i` seja um valor entre 1 e o comprimento de `w`.

b) Desenvolva em F, recorrendo ao módulo acima e assegurando a independência da implementação, uma subrotina `f` tal que `f(w,n)` recebe no primeiro argumento uma lista compactada `w` devolve no parâmetro de saída `n` o comprimento da maior sequência de números consecutivos repetidos da lista `w`. Por exemplo, no caso da lista anterior, o comprimento da maior sequência é 3, correspondente às três ocorrências consecutivas do número 14. Descreva o que se entende por independência da implementação e as suas vantagens.

### Grupo III

[1,0 + 2,0]

a) Defina efeito colateral e explique a sua importância no desenvolvimento de programas em F. Relacione a noção de efeito colateral com o desenvolvimento de funções e subrotinas em F.

b) Considere a função `g` seguinte, definida em F. Descreva a funcionalidade desta função. Defina uma versão recursiva desta função, com a mesma funcionalidade de `g`.

```
function g(v1,v2) result(r)
integer, dimension(:), intent(in) :: v1,v2
integer :: r, i, j

if (size(v1)==size(v2)) then
  i=(size(v1)/2)
  j=(size(v1)/2)+1
  r=0
  do
    if (i==0) then
      if(j==size(v1)) then
        r=r+v1(j)*v2(j)
      end if
      exit
    else
      r=r+v1(i)*v2(i)+v1(j)*v2(j)
      i=i-1
      j=j+1
    end if
  end do
end if
end function g
```