

Computação e Programação

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

Exame 1A

Departamento de Matemática, IST

23 de Janeiro de 2004, 9h00m

Duração: 2h30m

Grupo I

[1,5 + 1,5]

a) Desenvolva em MATLAB, sem recorrer à função predefinida `max`, uma função `f` que recebe como argumento um vector de números inteiros, percorre o vector dado recorrendo a um ciclo `while` e determina o número de elementos em posições pares que são iguais ao máximo do vector.

b) Desenvolva em MATLAB, sem recorrer à função predefinida `max`, 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 número de elementos em colunas pares que são iguais ao máximo da matriz.

Grupo II

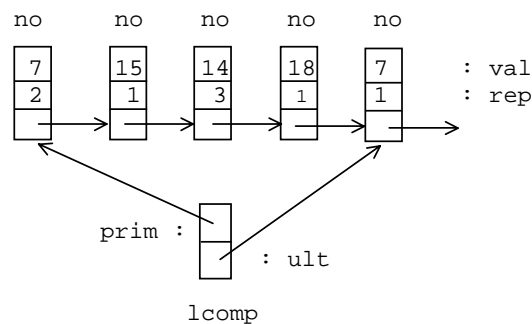
[2,0 + 2,0]

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 o número de vezes que este ocorre repetido. Por exemplo, a lista $\{7, 7, 15, 14, 14, 14, 18, 7\}$ é representada de forma compacta por $\{(7, 2), (15, 1), (14, 3), (18, 1), (7, 1)\}$. 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 :: rep
  type(no), pointer :: seg
end type no
```

a lista anterior 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 o número de vezes que ele aparece repetido;
- `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 o número de vezes que ele aparece repetido;
- `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) Por lista ordenada entende-se uma lista em que os elementos surgem por ordem crescente (em sentido lato, isto é podem surgir elementos consecutivos iguais). Por fusão de duas listas ordenadas entende-se a lista ordenada formada pelos elementos das duas listas. Desenvolva em F, recorrendo ao módulo acima, uma subrotina `f` tal que, no caso de `w1` e `w2` serem listas compactadas ordenadas, `f(w1,w2,w)` devolve no parâmetro de saída `w` a lista compactada que resulta da fusão das listas `w1` e `w2`.

Grupo III

[1,0 + 2,0]

- a) Descreva a estrutura geral de um módulo em F, indicando a utilidade de cada uma das suas partes.
- 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(v) result(r)
integer, dimension(:), intent(in) :: v
integer :: r, i, j

i=1
j=size(v)
r=0
do
  if (i>j) then
    exit
  elseif(i==j) then
    r=r+v(i)
    exit
  else
    r=r+v(i)+v(j)
    i=i+1
    j=j-1
  end if
end do
end function g
```