

Programação Imperativa

LMCC (1º ano)

Exame de Época Especial

Data: 7 de Setembro de 2005
Hora: 14:00

Dispõe de **2:00 horas** para realizar este exame.

Responda na folha fornecida em anexo
e não se esqueça de preencher o cabeçalho

1 Questão (cálculo numérico)

Especifique o algoritmo para redução de uma fracção à sua forma mais simples, i. e., o numerador é o mais pequeno possível. A função especificada deverá receber uma fracção (um par constituído por numerador e denominador) como argumento e dar como resultado uma nova fracção. Posteriormente codifique em C a sua solução.

%_____

2 Questão (ficheiros)

Escreva em C, um programa que lê um ficheiro de texto, e que no fim produz um resultado estatístico indicando quantas vezes ocorreu cada letra de A a Z (para o efeito considere que as ocorrências de a e de A são contabilizadas no mesmo contador) e qual a sua frequência no texto lido (a frequência de um carácter é a razão entre o número de vezes que esse carácter foi encontrado e o número total de caracteres encontrados no texto).

%_____

3 Questão (recursividade)

Desenvolva duas versões em C da função f , uma recursiva e outra iterativa. A função f recebe um inteiro positivo e cria uma lista ligada de inteiros entre 0 e o inteiro lido. Por exemplo, a invocação $f(5)$ daria origem à lista $[0, 1, 2, 3, 4, 5]$. A assinatura da função deverá ser a mesma e seguir a seguinte forma:

```
typedef struct sLint
{
    int n;
    struct sLint *seg;
} *Lint, Nodo;
```

...

```
Lint f(int num)
{
    ...
}
%-----
```

4 Questão (árvore binárias de procura)

Considere as seguintes definições para uma árvore binária de procura de números inteiros e para uma lista ligada de inteiros respectivamente:

```
/* Árvore Binária de Procura */
typedef struct sABPInt
{
    int valor;
    struct sABPInt *esq, *dir;
} NODO, *ABPint;

/* Lista Ligada */
typedef struct sLLInt
{
    int valor;
    struct sLLInt *seg;
} ELEM, Lista;
```

- a) Desenvolva a função em C que converte a árvore numa lista atravessando a árvore segundo o algoritmo de travessia *inorder*:

```
Lista convAL( ABPint a)
{
    ...
}
```

- b) Especifique a função, em C, que retorna o elemento maior armazenado numa árvore binária de procura.
- c) Especifique a função, em C, que recebendo duas lista ligadas de inteiros ordenadas dá como resulta uma lista ligada de inteiros ordenada resultante da fusão das duas listas passadas como argumento.

%-----

5 Questão (estrutura de dados)

Considere uma árvore genealógica ascendente (que relaciona uma pessoa com os seus pais) e que em termos mais formais pode ser descrita da seguinte maneira:

```
AG = nome * Pai * mãe | desconhecido
Pai = Mãe = AG
```

1. Especifique em C o tipo de dados AG;
2. Especifique a função *ascendentes*, que dada uma árvore genealógica e o nome de uma pessoa, imprime no *stdout* os nomes de todos os ascendentes conhecidos dessa pessoa.
3. Especifique uma função que conte o número de pessoas incluídas numa determinada AG.

%-----