

Programação Imperativa – LMCC (1º ano)
Paradigmas da Programação II – LESI (1º ano)

Exame de 1ª Chamada

Data: 21 de Junho de 2006
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

Questão 1 (cálculo numérico)

Considere o seguinte tipo de dados que pode ser usado para representar fracções:

```
typedef struct sFrac
{ int numerador;
  int denominador;
} Frac;
```

Considere ainda os tipos criados para suportar o armazenamento de listas de fracções:

```
typedef struct sLFrac
{ Frac f;
  struct sLFrac *seg;
} NodoListaFrac, *ListaFrac;
```

Defina em C uma função de nome *somaListaFrac* que recebe uma lista de fracções e produz como resultado uma fracção resultante da sua soma.

```
Frac somaListaFrac( ListaFrac lf )
{
    ...
}
```

%_____

Questão 2 (strings em C)

Leia do teclado uma sequência de linhas (cada uma não excede o comprimento de 256 caracteres e a última começa pelo caracter "."). Por cada linha lida indique quantas vezes e em que posições ocorre a substring "ana" (considere que o primeiro caracter está na posição 0).

A título de exemplo, repare que em "banana" há 2 ocorrências nas posições 1 e 3, e que em "ananas" a substring "ana" ocorre também 2 vezes, nas posições 0 e 2.

%_____

Questão 3 (recursividade)

Considere a seguinte definição da multiplicação de dois números inteiros e positivos que tira partido do facto de que a divisão e multiplicação por 2 serem operações muito eficientes.

$$x \times y = \begin{cases} 0 & \text{Se } y = 0 \\ (x * 2) \times y' & \text{Se } y = 2 * y' \\ (x * 2) \times y' + x & \text{Se } y = 2 * y' + 1 \end{cases}$$

Use esta definição para definir uma função de multiplicação de números inteiros. Defina duas versões de tal função: uma recursiva e outra iterativa.

%_____

Questão 4 (arrays e produto cartesiano em C)

No Concurso Nacional de Programação Lógica e Funcional, CeNPLf, há 2 provas (uma de manhã e outra de tarde) com 5 problemas cada. Cada problema é classificado de 0 a 100 e a pontuação final de uma equipa (com 2 concorrentes) é o máximo entre a pontuação de cada uma das provas (somatório dos pontos de cada problema).

Para armazenar a informação relativa a cada equipa, definiu-se o seguinte tipo em C:

```
typedef char String[30];
typedef struct E{
    int classifs[provas][probs];
    int classFinal;
    String concorr1, concorr2;
} tEquipa;
```

Responda então às alíneas seguintes:

- Diga o que terão de ser os identificadores `provas` e `probs` e como teriam de ser declarados para que a definição de `tEquipa`, acima, compile sem erros.
- Diga se a declaração de `String`, acima, poderia ser substituída pela seguinte:

```
typedef char *String;
```

e, em caso afirmativo, diga se algo mudaria em termos do código C que trabalha com os campos `concorr1` e `concorr2`.

- Defina o tipo `tEquipas` para guardar num array a informação relativa às 25 equipas concorrentes.
- Desenvolva em C a função `tEquipa calcFinal(tEquipa e)` que calcula a classificação final da equipa `e`, recebida como parâmetro, e a devolve com o campo `classFinal` preenchido.
- Desenvolva em C a função `float média(tEquipa e, int prova)` que calcula e devolve a classificação média obtida pela equipa `e` na prova da manhã ou da tarde.

%_____