

# Paradigmas da Programação II

## LESI + LMCC (1º ano)

Exame de 1ª Época — 2ª Chamada

Data: 08 de Julho de 2004

Hora: 14:00

Versão:   A  

Dispõe de **2:30 horas** para realizar este exame,  
que tem 7 perguntas V/F (14valores) e 3 perguntas de desenvolvimento (6valores).

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

PARA CADA ALÍNEA,  
INDIQUE SE É (F)alsa ou (V)erdadeira,

Critério de Correção:

4 alíneas certas ⇒ **100%**

3 alíneas certas ⇒ **50%**

2 alíneas certas ⇒ **25%**

### I.PARTE: Questionário V/F

#### 1 Questão (proc. numérico sequencial)

% \_\_\_\_\_

#### 2 Questão (arrays e proc. numérico sequencial)

Observe a função fun() escrita na linguagem C como se lista abaixo:

```
int fun(int v[])
{ int i,s;
  for ( s=0,i=1; i<v[0]; i++ ) { s += v[i]; }
  return( s );
}
```

e em relação a ela, analise a veracidade das afirmações seguintes:

a) A função calcula o somatório de todos elementos de um vector.

- b) Se o valor contido na primeira posição do vector for 10, a função retorna o somatório dos 10 primeiros elementos do vector a seguir ao primeiro.
- c) Porque o cabeçalho da função está bem escrito, pode afirmar-se que os *arrays* em C podem ser declarados com comprimento variável, desconhecido na hora da compilação.
- d) Qualquer array de inteiros pode ser passado como argumento de `fun()`, independentemente do seu tamanho.

%\_\_\_\_\_

### 3 Questão (strings)

%\_\_\_\_\_

### 4 Questão (matrizes)

%\_\_\_\_\_

### 5 Questão (ficheiros)

No exame anterior discutiu-se o uso do tipo `tHorario` para armazenar um horário relativo à ocupação de uma sala (nome da disciplina), das 8h00 às 20h00 e de 2<sup>a</sup>f a Sábado, sendo:

```
#define MaxH 12
#define MaxD 6
#define MaxS 30
typedef char tDisc[MaxS];
typedef tDisc tHorario[MaxH][MaxD];
```

Agora pretende-se analisar a sua preservação em ficheiro. Recorde, então, os seus conhecimentos sobre ficheiros em C e analise a veracidade das afirmações seguintes:

- a) Era possível guardar o horário H, do tipo indicado, num ficheiro `out`, usando a seguinte instrução:

```
fwrite(H,1,sizeof(tHorario),out);
```

sendo o ficheiro previamente aberto como

```
out = fopen("sala1.bin","a");
```

- b) Era possível guardar o horário H, do tipo indicado, num ficheiro `out`, usando o seguinte ciclo:

```
for(j=0;j<MaxD;j++)
  {for(i=0;i<MaxH;i++)
    {fwrite(H[i][j],1,sizeof(tDisc),out);}}
```

sendo o ficheiro previamente aberto como

```
out = fopen("sala1.bin","w");
```

- c) Era possível guardar o horário H, do tipo indicado, num ficheiro `out`, usando o seguinte ciclo:

```
for(i=0;i<MaxH;i++)
  {for(j=0;j<MaxD;j++)
    {fprintf(out,"%s ",H[i][j]);} fprintf(out,"\n");}
```

sendo possível ler e editar, fora do programa, o ficheiro produzido.

d) Era possível guardar o horário H, do tipo indicado, num ficheiro out, usando o seguinte ciclo:

```
for(i=0;i<MaxH;i++)
  {for(j=0;j<MaxD;j++)
    {fprintf(out,"%s ",H[i][j]);} fprintf(out,"\n");}
```

ficando o ficheiro produzido com um total de 72 linhas.

%\_\_\_\_\_

## 6 Questão (polinómios)

%\_\_\_\_\_

## 7 Questão (listas)

%\_\_\_\_\_

## II.PARTE: Perguntas de Desenvolvimento

## 8 Questão (função simples)

%\_\_\_\_\_

## 9 Questão (função complexa)

Supondo que as palavras-chave de um dicionário estão armazenadas numa lista generalizada, tal como sugerido nas aulas práticas, de tal modo que cada letra só se escreve um vez sendo partilhada por todas as palavras que começam com o mesmo prefixo, escreva uma função que faça a listagem de todas as chaves do dicionário por ordem alfabética. Assuma que cada nodo da lista generalizada é do tipo `tNodo` e que o dicionário é um apontador para o início da lista do tipo `tLista`, assim declarados

```
typedef struct N
  { char Letra;
    struct N *LetraSeg;
    struct N *LetraAlt;
  } tNodo, *tLista;
```

%\_\_\_\_\_

## 10 Questão (estrutura de dados)

%\_\_\_\_\_