

Paradigmas da Programação II

LESI + LMCC (1º ano)

Exame de 2ª Época

Data: 20 de Julho de 2004

Hora: 09:30

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, <u>Critério de Correção:</u> 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)

% _____

3 Questão (strings)

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

```
bool misterio(char pal[])
{ int i,j;
  i=0; j=strlen(pal)-1;
  while ( (pal[i]==pal[j])&&(i<j) ) { i++; j--; }
  return( i>=j );
```

```
}
```

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

- Se a função for invocada da seguinte forma `misterio("pedro");`, retorna "falso".
- Se a função for invocada da seguinte forma `misterio("pedro");`, não retorna nada pois dará um erro de execução pelo seguinte motivo: *encontrado um texto constante quando se esperava uma array de caracteres*.
- O erro referido na alínea anterior, não chega sequer a ser um erro de execução pois seria detectado e rejeitado pelo compilador.
- A função `misterio2()`, escrita abaixo na linguagem C, é equivalente à anterior `misterio()`, embora muito menos eficiente:

```
bool misterio2(char pal[])
{ int i;
  char *pi;
  pi=(char *)malloc(strlen(pal));
  i(strlen(pal)-1;
  while ( pal[i] ) { *pi=pal[i--]; pi++;}
  *pi='\0';
  return( strcmp(pal,pi)==0 );
}
```

%_____

4 Questão (matrizes)

%_____

5 Questão (ficheiros)

%_____

6 Questão (polinómios)

Nas aulas práticas discutiu-se com detalhe diferentes maneiras de armazenar polinómios. Recorde-as, então, e em relação a elas analise a veracidade das afirmações seguintes:

- A função abaixo subtrai dois polinómios de grau menor que G se eles estiverem armazenados num array por ordem crescente do expoente de cada monómio:

```
void misterioP1(tPoli1 A, tPoli1 B, tPoli1 S)
{ int i=0;
  while ( i<G ) { S[i]=A[i]-B[i]; i++;}
}
```

- Uma declaração adequada para `tPoli1`, usado na alínea anterior, seria:

```
#define G 12
typedef struct M1
{ int exp;
  float coef;
} tMono1;
typedef tMono1 tPoli1[G];
```

- c) Como alternativa à primeira forma de armazenamento (*em array*), o armazenamento em lista pode ser feito com as seguintes declarações:

```
#define G 12
typedef struct M2
    { int exp;
      float coef;
    } tMono2;
typedef struct P2
    { tMono2 mono;
      struct P2 *seg;
    } *Poli2;
```

sendo recomendável quando os polinómios a manipular são de grau elevado mas muito incompletos.

- d) Usando listas ligadas de monómios (tipo `tPoli3`) para armazenar os termos de um polinómio por ordem decrescente de grau, a função abaixo pode ser usada para multiplicar um polinómio por um escalar:

```
void misterioP3(float s, tPoli3 A)
{
    while ( A ) { A->coef *= s; A=A->seg;}
}
```

%_____

7 Questão (listas)

%_____

II.PARTE: Perguntas de Desenvolvimento

8 Questão (função simples)

%_____

9 Questão (função complexa)

%_____

10 Questão (estrutura de dados)

Uma família é formada por um **casal** (**Mãe** e **Pai**, em que qualquer um dos dois **progenitores** pode ser "desconhecido") e uma lista de **Filhos**. Mãe, Pai e Filho são **peçoas** (caracterizadas por um nome próprio e apelido, data e local de nascimento, data e local de óbito e identificadas univocamente por um número de registo), sendo que cada pessoa além de filho, pode ser progenitor e pode fazer parte de mais do que um casal.

Declare uma Estrutura de dados conveniente para armazenar todas as pessoas registadas numa época numa determinada Conservatória e as famílias em que essas pessoas participaram e que foram encontradas com base nos registos de casamento do qual também se conhece a data e o local de matrimónio.

A estrutura concebida deve facilitar a listagem, ordenada e sem repetições, de todas as pessoas registadas e a listagem de todas as famílias por ordem crescente de data de casamento. %_____