

# Programação Imperativa

## 1º Ano – LEI/LCC

Exame de Recurso – Duração: 2 horas

21 de Julho de 2007

### Parte I

Esta parte do exame representa 12 valores da cotação total. Cada uma das 6 alíneas está cotada em 2 valores. **A obtenção de uma classificação abaixo de 8 valores nesta parte implica a reprovação no exame.**

1. Defina uma função `void reverse (int a [N], int b[N])` que coloca no *array* `b` os `N` elementos do *array* `a` por ordem inversa.
2. Refaça o exercício anterior mas considere agora que a função recebe apenas o primeiro argumento e que retorna uma lista ligada com os elementos do array por ordem inversa. Use para isso o seguinte tipo para representar listas ligadas de inteiros.

```
typedef struct slista {
    int valor;
    struct slista *seg;
} *Lista;
```

3. Pretende-se calcular a nota prática a PI obtida através das 3 provas de avaliação. Esta nota obtém-se fazendo a média entre a nota do 3º teste e a melhor dos 2 primeiros testes.

Complete a definição da função que recebe um array com as notas dos 3 testes e devolve um array com a nota prática final.

```
typedef struct snotast{
    int numaluno;
    int nota1, nota2, nota3
} NotasTestes;
```

```
typedef struct snotasp{
    int numaluno;
    int nota;
} NotasP;
```

```
int notaspraticas(NotasTestes nt[MAX], NotasP npf[MAX])
{ ...
return 0;
}
```

4. Defina uma função `int nLinhas (char *)` que calcula o número de linhas de um ficheiro de texto (se o ficheiro estiver vazio ou não existir, a função deve retornar um valor negativo).
5. Relembre o *jogo da forca* implementado no trabalho prático.
  - (a) Defina uma função `int ocorre (char c, char *s)` que dado um caracter `c` determina se ele ocorre numa dada *string* `s`.
  - (b) Defina uma função `void mostra (char *desafio, char *letras)` que, escreve no ecran a *string* `desafio`, ocultando (com o caracter `'*'`) todas os caracteres que **não** ocorram na *string* `letras`. Poderá usar a função definida na alínea anterior (mesmo se não tiver respondido a essa alínea).

## Parte II

1. Relembre o popular jogo de SUDOKU, em que, num tabuleiro de  $9 \times 9$  posições (divididos em 9 quadrados de  $3 \times 3$  cada) se devem dispor algarismos de 1 a 9 de forma a que:

- cada algarismo aparece uma única vez em cada linha
  - cada algarismo aparece uma única vez em cada coluna
  - cada algarismo aparece uma única vez em cada um dos 9 quadrados interiores
- (a) Defina uma função que `int linhaOK (int l[9])` que, dado um vector com 9 inteiros, retorna verdadeiro se esse nesse array aparecerem os números de 1 a 9.
- (b) Defina uma função `void coluna (int tabuleiro [9][9], int c, int col [9])` que recebe um tabuleiro, um índice de coluna e preenche um vector `col` com os valores que estão na coluna `c` do `tabuleiro`.
- (c) Defina uma função `void quadrado (int tabuleiro [9][9], int q, int quad [9])` que recebe um tabuleiro, um índice de quadrado (considere os quadrados numerados de 1 a 9 começando no canto superior esquerdo e terminando no canto inferior direito) e preenche um vector `quad` com os valores que estão no quadrado `q` do `tabuleiro`.

Com as funções definidas atrás podemos definir uma função que testa se um dado tabuleiro está bem preenchido:

```
int sudokuOK (int tabuleiro [9][9]) {
    int aux [9], i, r = 1;

    // teste das linhas
    for (i=0; ((i<9) && r); i++)
        r = linhaOK(tabuleiro [i]);
    // teste das colunas
    for (i=0; ((i<9) && r); i++) {
        coluna (tabuleiro, i, aux); r = linhaOK (aux);
    }
    // teste dos quadrados
    for (i=0; ((i<9) && r); i++) {
        quadrado (tabuleiro, i, aux); r = linhaOK (aux);
    }
    return r;
}
```

- (d) Considere agora que usamos o número -1 para assinalar que uma dada posição do tabuleiro ainda não está preenchida. Pretende-se por isso definir uma função que dado um tabuleiro não totalmente preenchido, retorne a lista de posições para as quais existe uma única alternativa de preenchimento imediato (porque todas as outras alternativas colidem quer com os elementos que estão nessa linha, coluna ou quadrado interior. Defina então essa função, começando por definir um tipo de dados apropriado para retornar esta informação.