

# Programação Imperativa

## 1º Ano – LEI/LCC

Teste – Duração: 2 horas

27 de Junho de 2008 (Salas 2201 a 2204)

### Parte I

Esta parte do teste 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 teste.**

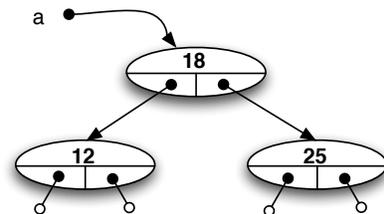
1. Defina uma função `int conta (char c, char *s)` que conta o número de ocorrências do caracter `c` na *string* `s` (terminada com o caracter nulo).
2. Seja `N` uma constante definida (usando a directiva `#define N...`). Defina uma função `void linha (int m [N] [N], int i, int l [N])` que copia para `l` o conteúdo da linha `i` da matriz `m` (linha,coluna).
3. Considere a seguinte definição para implementar listas ligadas de inteiros.

```
typedef struct celula *LInt;  
struct celula {  
    int valor;  
    LInt next;  
};
```

Defina uma função `LInt insert_at (LInt, int, int)` que dada uma lista de inteiros, um inteiro e uma posição, insere esse elemento na lista, nessa posição. Por exemplo, a instrução `x = insert_at (x, 10, 0)` insere o número 10 no início da lista. Se a posição fornecida for maior do que o comprimento da lista, essa far-se-á no final.

4. Considere a seguinte definição para árvores binárias de procura.

```
typedef struct nodo *ABPInt;  
struct nodo {  
    int valor;  
    ABPInt esq, dir;  
};
```



- (a) Apresente as instruções necessárias para a construção da árvore apresentada na figura acima e a sua atribuição à variável `a` do tipo `ABPInt`.
  - (b) Apresente uma definição de uma função `int menor (ABPInt)` que receba uma árvore de procura não vazia e calcule o menor elemento dessa árvore (sem percorrer necessariamente toda a árvore).
5. Defina uma função `int llongline (char *)` que, dado o nome de um ficheiro de texto, devolva o tamanho da linha mais longa desse ficheiro. A função deve retornar `-1` caso o ficheiro não exista ou não seja possível consultá-lo.

## Parte II

1. Considere a seguinte definição da função `strcmp`

```
int strcmp (char *a, char *b) {
while ((*a) && (*b) && (*a == *b)) {
    a++; b++;
}
return (*a - *b);
}
```

Rescreva-a de forma a usar os dois parâmetros como vectores de caracteres, em vez de, tal como na definição apresentada, endereços de memória.

```
int strcmp (char a[], char b[]) {
...
}
```

2. Defina uma função `int capitaliza (char *)` que, dado o nome de um ficheiro de texto, escreve no écran (*stdout*) esse ficheiro com a primeira letra de cada frase transformada na correspondente maiúscula. Note que o início de uma frase se detecta como o primeiro carácter não branco após um ponto final.
3. Relembre a definição de árvores binárias de procura da alínea 4. da Parte I. Defina uma função `int proximo (ABPInt, int, int*)` que, dada uma árvore e um inteiro, coloca no endereço (passado como 3º parâmetro) o valor do menor elemento da árvore que é maior do que esse inteiro. A função deve devolver 0 se tal não for possível (i.e., o inteiro dado é maior ou igual a todos os elementos da árvore) e 1 nos outros casos. No exemplo da referida alínea, ao executarmos a instrução

```
x = proximo (a,13,&y);
```

deve colocar nas variáveis `x` e `y` os valores 1 e 18, respectivamente.

**Note que** a definição pretendida deve ter em linha de conta que a árvore é de procura e por isso não deve percorrer toda a árvore.