

# Programação Imperativa – Exame de Recurso

25 de Junho de 2016 – Duração: 2h00m

## Parte A

1. Apresente uma definição da função pré-definida em C `char *strcpy (char *dest, char source[])` que copia a string `source` para `dest` retornando o valor desta última.
2. Defina uma função `void strnoV (char s[])` que retira todas as vogais de uma string.
3. Defina uma função `int dumpABin (ABin a, int v[], int N)` que dada uma árvore `a`, preenche o array `v` com os elementos da árvore segundo uma travessia `inorder`. A função deverá preencher no máximo `N` elementos e retornar o número de elementos preenchidos.  

```
typedef struct nodo {  
    int valor;  
    struct nodo *esq, *dir;  
} *ABin;
```
4. Apresente uma definição não recursiva da função `int lookupAB (ABin a, int x)` que testa se um elemento pertence a uma árvore binária de procura.

## Parte B

Considere que, para contar as palavras de um texto, se faz uso de uma estrutura baseada numa lista ligada que associa a cada palavra o respectivo número de ocorrências. Para tal, considere as declarações de tipos à direita.

```
typedef struct listaP{  
    char *pal;  
    int cont;  
    struct listaP *prox;  
} Nodo, *Hist;
```

1. Defina a função `int inc(Hist *h, char *pal)` que incrementa o número de ocorrências da palavra `pal` (retornando o novo valor). Considere para o efeito que a lista mantém as palavras ordenadas lexicograficamente. Note que se a palavra não existir na lista, deve ser criada uma nova entrada.
2. Defina a função `char *remMaisFreq(Hist *h, int *count)` que, dado um histograma `h`, remova a entrada correspondente à palavra mais frequente, retornando-a e armazenando a sua contagem na posição de memória apontada por `count`.
3. Escreva um programa C que, utilizando os tipos e as funções definidas, leia um texto (de `stdin`) e imprima (em `stdout`) as 10 palavras mais frequentes com mais de três caracteres (e a respectiva contagem). Deve atender aos seguintes aspectos:
  - Pode considerar que o tamanho máximo das palavras que ocorrem no texto é 60;
  - As palavras devem passar por um processo de filtragem onde se removem todos os caracteres que não letras e se convertem para maiúsculas (e.g. "gato22,", "GaTo" ou "Gato,,," devem ser consideradas a mesma palavra: "GATO").
  - Toda a memória alocada pelo programa deve explicitamente libertada.
  - Utilize as funções definidas nas bibliotecas standard do C (em particular as de manipulação de *strings* e caracteres disponibilizadas em `string.h` e `ctype.h`).