

17 de Junho de 2015

Parte A

Considere as seguintes definições de tipos:

```
typedef struct slist {
    int valor;
    struct slist *prox;
} *LInt;
```

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

1. Apresente uma definição da função pré-definida em C `char *strstr (char s1[], char s2[])` que determina a posição onde a string `s2` ocorre em `s1`. A função deverá retornar `NULL` caso `s2` não ocorra em `s1`.
2. Defina uma função `int maxInd (int v[], int n)` que, dado um vector `v` com `n` inteiros, retorna o índice do maior elemento do vector.
3. Apresente uma definição da função `void concat (LInt *a, LInt b)` que acrescenta a lista `b` a `*a`.
4. Defina uma função `LInt nivel (ABin a, int n)` que, dada uma árvore binária, constrói uma lista com os valores dos elementos que estão armazenados na árvore ao nível `n` (assuma que a raiz da árvore está ao nível 1).

Parte B

1. Escreva um programa que, usando entre outras a função `maxInd` referida atrás, lê uma sequência de no máximo 100 números inteiros sem repetições e imprime qual o maior número lido, bem como o seu número de ordem.
Note que a fase de leitura deverá terminar quando forem atingidos os 100 números lidos **ou** quando for lido um número que já tenha sido lido.
2. Considere o seguinte tipo de dados para representar listas de strings:

```
typedef struct strlist {
    char *string;
    struct strlist *prox;
} *StrList;
```

- (a) Defina uma função `void unwords (StrList s, char t[], int N)` que, dada uma lista de palavras, preenche o array `t` com as palavras dessa lista, separadas por um espaço. A função deve preencher no máximo `N` caracteres do array `t` e garantir que o último carácter preenchido é o carácter `'\0'`.
- (b) Defina uma função `int words (char t[], StrList *l)` que dada uma string `t` (terminada com o carácter `'\0'`) contendo várias palavras (as palavras são separadas por um ou mais espaços), coloca em `*l` a lista de strings com as várias palavras desse texto. A função retorna o número de palavras encontradas.
Os endereços contidos nessa lista devem corresponder a endereços dentro da string de entrada e por isso não deverá ser alocado espaço adicional para armazenar essas palavras.

3. Defina a função `int maxuniqueseq (int v[], int N)` que, dado um array `v` com `N` inteiros, determina o tamanho do maior sub-array de `v` sem elementos repetidos.
- Por exemplo, se o array contiver os (10) elementos `[8, 2, 2, 3, 4, 1, 6, 5, 1, 7]`, por essa ordem, a função deverá retornar 6 (correspondendo ao sub-array `[2, 3, 4, 1, 6, 5]`).