

Programação Imperativa

Exame de Recurso

1º Ano – LEI/LCC

9 de Julho de 2014 (Duração: 2:30 horas)

1. Responda às seguintes questões sobre strings.

- (a) Uma palavra diz-se *palíndrome* se se ler da mesma forma da esquerda para a direita ou da direita para a esquerda. Por exemplo, “sugus” é palíndrome. Defina uma função `int palindrome (char s[])` que testa se uma string é palíndrome.
- (b) Defina a função `char *strlchr (char *s, char c)` que localiza a **última** ocorrência do carácter `c` dentro da string `s`.

A função deverá retornar o endereço da posição onde a última ocorrência do carácter foi encontrada, ou NULL caso o carácter não tenha sido encontrado.

2. Defina uma função `void ordena (int nums [], int notas [], int n)` que recebe dois arrays com os números e notas dos alunos de uma turma e ordena os arrays por ordem decrescente da nota.

Por exemplo, se o array de notas contém {10, 14, 11, 12} e o de números contém {1024, 3096, 5087, 6178}, após a invocação desta função deverão passar a conter {14, 12, 11, 10} e {3098, 6178, 5087, 1024}, respectivamente.

3. Considere o seguinte tipo para representar listas ligadas de inteiros:

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

- (a) Defina uma função `int dumpL (LInt l, int v[], int N)` que, dada uma lista `l`, preenche o array `v` com os elementos da lista. A função deverá preencher no máximo `N` elementos e retornar o número de elementos preenchidos.
- (b) Defina uma função `LInt somas (LInt l)` que, dada uma lista de inteiros, constrói uma nova lista de inteiros contendo as somas acumuladas da lista original (que deverá permanecer inalterada).

Por exemplo, se a lista `l` tiver os valores [1,2,3,4] a lista contruída pela invocação de `somas (l)` deverá conter os valores [1,3,6,10].

- (c) Defina uma função `void remreps (LInt l)` que, dada uma lista ordenada de inteiros, elimina dessa lista todos os valores repetidos assegurando que o espaço de memória correspondente aos nós removidos é correctamente libertado.
4. Considere o seguinte tipo para representar árvores binárias de inteiros:

```
typedef struct sbin{
    int valor;
    struct sbin *esq, *dir;
} *Abin;
```

Apresente uma definição da função `int contaFolhas (Abin a)` que dada uma árvore binária de inteiros, conta quantos dos seus nodos são folhas, i.e., que não têm nenhum descendente.

5. O problema das N-raíñas consiste em, num tabuleiro de xadrez de dimensão N, colocar N raíñas de forma a que nenhuma ataque a outra.

Uma raíña na posição (a, b) ataca outra na posição (c, d) se alguma das seguintes condições se verificar:

- estiverem na mesma linha (i.e., $b = d$),
- estiverem na mesma coluna (i.e., $a = c$),
- estiverem na mesma diagonal (i.e., $a + b = c + d$ ou $a - b = c - d$).

Apresente uma definição da função `int validSol (int board[N] [N])` que, dada uma matriz descrevendo a posição de N raíñas num tabuleiro, retorna 1 se essa disposição constituir uma solução válida para o problema e 0 caso contrário.

Considere que `board[i] [j]` terá o valor 1, caso esteja colocada uma raíña na posição coluna i/linha j e 0 caso essa posição esteja vazia.