

## MC102—Algoritmos e Programação de Computadores

Turmas C e D

Segundo Semestre de 2010

## Lista de Exercícios 6

## Apontadores

1. Se `i` é uma variável e `p` aponta para `i`, qual(is) das seguintes expressões é(são) equivalente(s) a `i`? [1]

(a) `*p` (b) `&p` (c) `*&p` (d) `&*p` (e) `*i` (f) `&i` (g) `*&i` (h) `&*i`

2. Complete a função [1]:

```
void acheOsDoisMaiores(int a[], int n, int *maior, int *seg_maior);
```

3. Suponha as seguintes declarações [1]:

```
int a[] = {5, 15, 34, 54, 14, 2, 52, 72};  
int *p = &a[1], *q = &a[5];
```

- (a) Qual é o valor de `*(p+3)`?
- (b) Qual é o valor de `*(q-3)`?
- (c) Qual é o valor de `q-p`?
- (d) A condição `p < q` é verdadeira?
- (e) A condição `*p < *q` é verdadeira?

## Lista Ligada

Para os exercícios com listas ligadas, considere que um nó tem a estrutura usada em aula [*valor* e *apontador para o próximo nó*].

1. Mostre o passo a passo de uma inserção de um nó em uma lista ligada ordenada. Considere que o nó a ser inserido possui valor 10 e que a lista tem atualmente os seguintes valores: 1 2 4 7 9 13 14.
2. Mostre o passo a passo de uma operação de inserção de um novo nó em uma lista ligada ordenada em ordem decrescente. Depois disso, faça o programa correspondente.

3. Mostre o passo a passo de uma operação de remoção de um nó de uma lista ligada ordenada em ordem decrescente. Depois disso, faça o programa correspondente.
4. Dada uma lista ligada com  $n$  nós, mostre o passo a passo da separação desta lista em duas sub-listas. A primeira delas guardará os  $n/2$  primeiros nós e a segunda sub-lista guardará os  $n/2$  nós restantes. A primeira sub-lista deve ser apontada por  $p1$  e a segunda por  $p2$ . Ao final, imprima separadamente as listas  $p1$  e  $p2$ . Considere que  $n$  é dado como entrada. Depois disso, faça o programa que implementa esta operação.
5. Faça um programa que remova nós com valores pares de uma lista não ordenada. A lista final deverá conter apenas nós com valores ímpares.
6. Qual a diferença entre uma pilha e uma fila?
7. Complete a função abaixo de forma que ela inverta a lista (considere que `MeuRegistro` possui dois campos, um do tipo `int` e outro apontador para `MeuRegistro`):  

```
void inverteLista (MeuRegistro **inicioDaLista );
```

## Referências

- [1] K.N. King, *C programming: a modern approach*, second edition, W.W. Norton (2008).