

## Prova 2 – 05/11/2014

Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_

As respostas (texto ou códigos) devem ser escritos nas folhas resposta, **a caneta azul ou preta**. Quaisquer texto escrito em lápis será considerado rascunho. **Escreva seu nome em cada folha da prova.**

As Questões 1 e 2 são baseadas na listagem abaixo, que apresenta estruturas de dados para a manipulação de uma lista duplamente encadeada.

```
1 typedef struct elem{
2     int valor;
3     struct elem * ant; //elemento anterior
4     struct elem * prox; //elemento posterior
5 }Elem;
6 typedef struct lista{
7     Elem * primeiro; //primeiro elemento da lista encadeada
8     Elem * ultimo; //ultimo elemento da lista encadeada
9 }Lista;
```

⇒ **Questão 01 (2,5pt)** Crie uma função que remova o primeiro elemento ou o último elemento da lista. **Obs.** O código deve possuir todas as verificações e controles necessários para o uso correto dos ponteiros. O protótipo da função deverá ser:

```
1 void remove(Lista * lista, int prim);
```

O primeiro parâmetro é a lista onde ocorrerá a operação. O segundo parâmetro serve para indicar se o elemento a ser removido é o primeiro ou o último.

⇒ **Questão 02 (2,5pt)** Crie uma função que adicione um elemento na lista, de forma que os elementos fiquem ordenados por valor, e os números pares apareçam antes dos números ímpares. O protótipo da função deverá ser:

```
1 Elem * adiciona (Lista * lista, int valor);
```

- Exemplo de lista já ordenada: 10 20 30 40 5 15 23 25

As Questões 3 e 4 são baseadas no código abaixo, que apresenta a interface para um tipo abstrato Pilha. Nas questões abaixo, utilize apenas as funções declaradas.

```
1 typedef struct pilha Pilha;
2 Pilha* pilha_cria(void);
3 void pilha_push(Pilha* p, int v);
4 int pilha_pop(Pilha* p);
5 int pilha_vazia(Pilha* p);
6 void pilha_libera(Pilha* p);
```

⇒ **Questão 03 (2,5pt)** Crie uma função que remove todos os números pares de uma pilha. O protótipo da função deverá ser:

```
1 void pilha_remove_pares(Pilha* p);
```

- Entrada (do topo para o final): 1 3 8 10 11 9 4
- Retorno (do topo para o final): 1 3 11 9

⇒ **Questão 04 (2,5pt)** Assumindo que o tipo Pilha insere caracteres (`char`) ao invés de inteiros, crie uma função que verifica se uma cadeia de caracteres é um palíndromo, ou seja, podem ser lida da esquerda para direita ou da direita para esquerda. Utilize o tipo abstrato Pilha para resolver este problema. O protótipo da função deverá ser:

```
1 bool testa_palindromo(char* palavra);
```

- Entrada: “sopapos”
- Retorno: true