

## Prova 1 – 11/10/2016

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

As respostas (texto ou códigos) devem ser escritas nas folhas resposta. Pode-se usar lápis.

As Questões 2 e 3 da prova são baseadas nas listagens abaixo, que apresentam estruturas de dados para a manipulação de listas encadeadas e pilhas, respectivamente.

```

1 typedef struct no{
2     int valor;
3     struct no* prox;
4 } No;
5
6 typedef struct lista{
7     No * ini;
8     No * fim;
9 }Lista;
```

```

1
2 typedef struct carta{
3     int cor; //0 para preta e 1 para vermelha
4     int valor;
5     struct carta * proxCarta;
6 }Carta;
7
8 typedef struct leque{
9     Carta * topo;
10 } Leque;
11
12
13 //coloca uma carta na pilha
14 void empilha(Leque * leque, Carta * carta);
15
16 //retira a carta da pilha
17 Carta * desempilha(Leque * leque);
18
19 //nao retira a carta, apenas recebe seu conteudo
20 Carta * espia(Leque * leque);
```

⇒ **Questão 01 (3,0pt)** Crie uma função que retorne um vetor contendo todos os valores pares existentes em um vetor de entrada. Caso o vetor de entrada não possua nenhum valor par, o retorno deverá ser nulo. O tamanho do vetor de pares deverá ser retornado pelo parâmetro novoTam.

O protótipo da função deverá ser:

```
1 int * vetorPar( int n, int * vetor, int * novoTam);
```

⇒ **Questão 02 (3,5pt)** Crie uma função que remova os elementos de uma lista caso seus valores sejam pares. Suponha que exista acesso direto aos campos do tipo Lista e do tipo No.

O protótipo da função deverá ser:

```
1 void removePares(Lista * l);
```

⇒ **Questão 03 (2,5pt)** O empilhamento de cartas faz parte do jogo chamado paciência. Suponha que exista uma variação do jogo em que uma carta só possa ser empilhada no leque se sua cor (vermelha ou preta) for diferente da carta que estiver no topo do leque. Se além de serem de cores opostas a nova carta tiver um número imediatamente anterior ao número da carta do topo, a nova carta não é adicionada e a carta do topo será removida. Implemente uma função que execute essas regras. O acesso ao TAD deve ser feito usando somente as funções que foram disponibilizadas.

O protótipo da função deverá ser:

```
1 void jogar( Leque * leque, Carta * novaCarta );
```

⇒ **Questão 04 (1,0pt)** Coleções de valores podem ser armazenadas na forma de vetores ou listas encadeadas. Fale a respeito das vantagens e desvantagens de cada uma dessas duas estratégias.