INE 5384 - Estruturas de Dados – Turma: 0332B – Semestre: 2002/2
Gabarito da Prova 1

Questão 1:

a)

```
Método insereNoInicio(objeto Object);
Inicio
        elem Elemento;

        elem ← NOVO Elemento(objeto, NULL);
        se inicio = NULL então fim ← elem;
        elem.prox ← inicio;
        inicio ← elem;
        fim.prox ← elem;
Fim;
```

b)

```
Método excluiNoFinal();
Inicio
        ant, ptr Elemento;

        se inicio = NULL então Exceção EstruturaVazia();
        se inicio = fim então
        inicio
                inicio ← NULL;
                fim ← NULL;
        fim
        senão   inicio
                        ant ← inicio;
                        ptr ← inicio.prox;
                        enquanto ptr <> fim faça
                        inicio
                                ant ← ptr;
                                ptr ← ptr.prox;
                        fim;
                        ant.prox ← inicio;
                        fim ← ant;
                fim;
Fim;
```

c)

```
Método excluiNoInicio();
Inicio
        se inicio = NULL então Exceção EstruturaVazia();
        se inicio = fim então
        inicio
                inicio ← NULL;
                fim ← NULL;
        fim
        senão inicio
                        fim.prox ← inicio.prox;
                        inicio ← inicio.prox;
                fim;
Fim;
```

Questão 2:

a)

```
Método empilheP1(objeto Object);
Inicio
        se topoP1 = topoP2 – 1 então Exceção EstruturaCheia();
        topoP1 ← topoP1 + 1;
        nroElementosP1 ← nroElementosP1 + 1;
        Vetor[topoP1] ← objeto;
Fim;
```

b)

```
Método desempilheP1() retorna Object;
Inicio
        se topoP1 = -1 então Exceção EstruturaVazia();
        topoP1 ← topoP1 - 1;
        nroElementosP1 ← nroElementosP1 - 1;
        retorna Vetor[topoP1 + 1];
Fim;
```

c)

```
Método empilheP2(objeto Object);
Inicio
        se topoP2 = topoP1 + 1 então Exceção EstruturaCheia();
        topoP2 ← topoP2 – 1;
        nroElementosP2 ← nroElementosP2 + 1;
        Vetor[topoP2] ← objeto;
Fim;
```

d)

```
Método desempilheP2() retorna Object;
Inicio
        se topoP2 = Vetor.lenght +1 então Exceção EstruturaVazia();
        topoP2 ← topoP2 + 1;
        nroElementosP2 ← nroElementosP2 - 1;
        retorna Vetor[topoP2 - 1];
Fim;
```

Questão 3:

```
Método TrocaNavios(armazem1 inteiro, armazem2 inteiro);
Inicio
        tipo inteiro;     /* indica qual é o tipo de troca a fazer no deque: */
                          /* caso 0: ambos saem pela Saída */
                          /* caso 1: ambos saem pela Entrada */
                          /* caso2: armazem1 sai pela Saída e
                                    armazem2 sai pela Entrada */
        aux Object;
        tamanhoDeque inteiro;
        pilha1, pilha2 PilhaEncadeada;

        /* verificação de exceções */
        se retNumDados() = 0 então Exceção EstruturaVazia();
        se armazem1 >= retNumDados() ou armazem2 > retNumDados() ou
        armazem1 >= armazem2
                                  então Exceção ArmazensInvalidos();

        /* teste dos casos para atribuição de tipo */
        se armazem1 <= retNumDados() DIV 2 e
        armazem2 <= retNumDados() DIV 2 então tipo ← 0
        senão se armazem1 > retNumDados() DIV 2 e
        armazem2 > retNumDados() DIV 2 então tipo ← 1
                senão tipo ← 2;

        /* tratamento de cada tipo de troca */
        pilha1 ← NOVO PilhaEncadeada();
        tamanhoDeque ← retNumDados();
        caso tipo

        0 : inicio
                repita
                        pilha1.empilhe(excluaNaSaida());
                        /* guarda em separado o armazem1  */
                        se pilha1.retNumDados() = armazem1 então
                                aux ← pilha1.retTopo();
                until pilha1.retNumDados() = armazem2;
                incluaNaSaida(aux); /* inclui primeiro armazem1 */
                aux ← pilha1.desempilhe(); /* guarda o armazem2 */
                /* repõe os navios no deque, colocando o armazem2 no lugar de
                   armazem1 */
                enquanto pilha1.retNumDados() > 0 faça
                        se pilha1.retNumDados() = armazem1 então
                        inicio
                                incluaNaSaida(aux);
                                aux ← pilha1.desempilhe();
                        fim
                        senão incluaNaSaida(pilha1.desempilhe());
            fim;
```

```
1 : inicio
        repita
                pilha1.empilhe(excluaNaEntrada());
                /* guarda em separado o armazem 2  */
                /* armazem 2 é o primeiro que sai neste caso */
                se pilha1.retNumDados() = tamanhoDeque – armazem2 + 1
                então  aux ← pilha1.retTopo();
        até pilha1.retNumDados() = tamanhoDeque – armazem1 + 1;
        incluaNaEntrada(aux); /* inclui primeiro o armazem2 */
        aux ← pilha1.desempilhe(); /* guarda o armazem1 */
        /* repõe os navios no deque, colocando o armazem1 no lugar de
           armazem2 */
        enquanto pilha1.retNumDados() > 0 faça
                se pilha1.retNumDados() = tamanhoDeque – armazem2 + 1
                então
                inicio
                        incluaNaEntrada(aux);
                        aux ← pilha1.desempilhe();
                fim
                senão incluaNaEntrada(pilha1.desempilhe());
    fim;


2:  inicio
        pilha2 ← NOVO PilhaEncadeada();
        /* empilha o lado do deque onde esta o armazem1 na pilha 1 */
        repita
                pilha1.empilhe(excluaNaSaida());
        até pilha1.retNumDados() = armazem1;
        /* empilha o lado do deque onde esta o armazem2 na pilha 2 */
        repita
                pilha2.empilhe(excluaNaEntrada());
        até pilha2.retNumDados() = tamanhoDeque – armazem2 + 1;
        /* troca os dois armazéns de posição no deque
           retirando-os dos topos das pilhas dos lados opostos */
        incluaNaEntrada(pilha1.desempilhe());
        incluaNaSaida(pilha2.desempilhe());
        /* recoloca os demais navios no deque nos seus respectivos lados */
        enquanto pilha1.retNumDados() > 0 faça
                incluaNaSaida(pilha1.desempilhe());
        enquanto pilha2.retNumDados() > 0 faça
                incluaNaEntrada(pilha2.desempilhe());
    fim;
Fim;
```