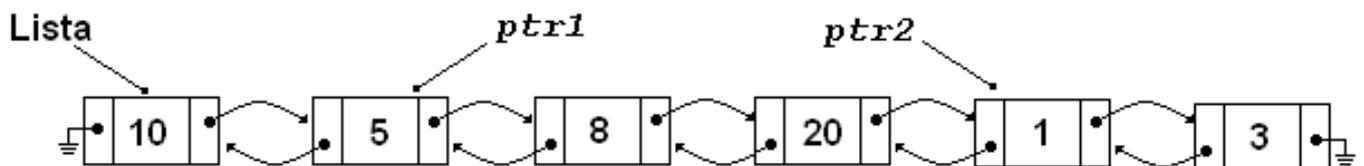


Nome do aluno: \_\_\_\_\_ Data: 13/03/2012  
 Resolva as questões abaixo identificando-as claramente na folha de respostas. Mantenha o silêncio na sala (mantendo desligado aparelhos eletrônicos). A interpretação das questões faz parte da prova. Horário limite: 15:40

(50 pontos) 1) Seja uma lista duplamente encadeada, com a estrutura e exemplo abaixo:

```
1 typedef struct _nodo {
2     struct _nodo *anterior;
3     struct _nodo *proximo;
4     int elemento;
5 } TipoNodo;
```



Sejam dados dois ponteiros para dois elementos distintos (*ptr1* e *ptr2*). Assumindo que:

- Estes ponteiros existem e não serão passados valores nulos;
- Que podem estar em qualquer posição da lista;
- *ptr1* vem “antes” na lista, de *ptr2*;
- Pode existir qualquer quantidade de elementos antes de *ptr1*, depois de *ptr2* e entre *ptr1* e *ptr2*.

Escreva uma função em C que recebe os ponteiros *ptr1* e *ptr2* e verifique a sanidade do caminho entre *ptr1* e *ptr2*. Ou seja se os passos entre *ptr1* e *ptr2* são o inverso dos passos entre *ptr2* e *ptr1*.

(25 pontos) 2) Durante o desenvolvimento de um sistema uma fila foi utilizada para armazenar os números referentes ao processamento, porém quando o processamento é encerrado é necessário que seja impresso na tela todos os itens dessa fila ao contrário. Implemente uma função `void imprime-fila-ao-contrario()`, que imprime na tela o conteúdo da fila ao contrário da ordem armazenada. Você pode utilizar apenas as funções de Fila para manipular os elementos, ou seja, **NÃO** pode acessar os elementos da fila diretamente. Segue o protótipo das funções que estão disponíveis e que podem ser utilizadas:

- `void enfilera(struct TipoFila *f)`
- `int desenfilera(struct TipoFila *f)`
- `int esta-vazia(struct TipoFila *f)`

Exemplo:

Imagine uma fila com os elementos armazenados:

10 20 -3 -15 20 50

Deve aparecer na tela:

50 20 -15 -3 20 10

**(10 pontos)** 3) Um novo aluno dos Pampas está com problemas ao desenvolver um programa em C que possui uma lista simplesmente encadeada. Por algum motivo a sua função `insere` não está de fato inserindo o elemento na lista. Você é a única pessoa que pode ajudá-lo com esse problema.

O código da função `insere` está abaixo. Fale sobre o problema e como resolvê-lo.

```
1 struct celula {
2     int conteudo;
3     struct celula *prox;
4 };
5
6 struct TipoLista {
7     struct celula *inicio;
8     int quantidade;
9 };
10
11 void insere-inicio( int x, struct TipoLista *l)
12 {
13     struct celula nova;
14     nova.conteudo = x;
15     nova.prox = l->inicio;
16     l->inicio = &nova;
17     (l->quantidade)++;
18 }
```

**(20 pontos)** 4) Considere a função recursiva abaixo. Escreva uma versão iterativa da função que simule o comportamento da versão recursiva. Considere as operações de pilha já implementadas.

```
1
2 int F( int *x, int n) {
3     if (n == 0)
4         return 0;
5     if (x[n] > 0)
6         return (x[n] + F(x, n-1));
7     else
8         return F(x, n-1);
9 }
```