

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. **Data:** 19/12/2013
Horário limite: 17:30

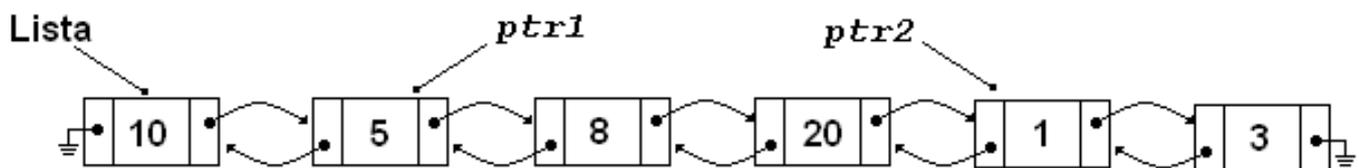
(10 pontos) 1) Qual o uso da função `scanf`? Descreva o que representa o valor de retorno dessa função.

(30 pontos) 2) Seja uma lista duplamente encadeada, com a estrutura e exemplo abaixo:

```

1 struct nodo {
2     struct nodo *anterior;
3     struct nodo *proximo;
4     int elemento;
5 };

```



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 rotacione a lista entre os ponteiros `ptr1` e `ptr2`, ou seja, deixe o caminho entre `ptr1` e `ptr2` o inverso, bem como o caminho de `ptr2` para `ptr1`. **ATENÇÃO:** a sua função deve apenas manipular os ponteiros sem copiar os valores (`int elemento`) e sem alocar novos nodos.

(30 pontos) 3) 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. Também a fila deve permanecer intacta após a execução da função `imprime-fila-ao-contrario()`.

Seguem os protótipos 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) 4) 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) 5) Suponha dada uma lista encadeada que armazena números inteiros. Cada célula da lista tem a estrutura abaixo.

<pre>1 struct cel { 2 int conteudo; 3 struct cel *prox; 4 };</pre>	<pre>1 struct TipoLista{ 2 int qtd; 3 struct cel *inicio; 4 };</pre>
--	--

Escreva uma função que transforme a lista em duas: a primeira contendo as células cujo conteúdo é par e a segunda aquelas cujo conteúdo é ímpar.

ATENÇÃO: A sua função **NÃO** deve alocar novos elementos, apenas manipular ponteiros.