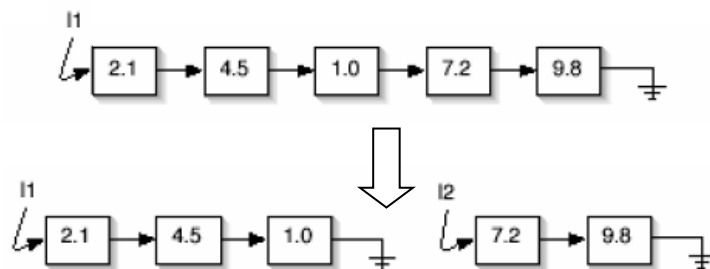




**Exercícios:**

1. Escreva uma versão da função busca para listas simplesmente encadeada. Defina também a estrutura de dados que você irá usar.
2. Dada uma lista simplesmente encadeada qualquer, como exemplificado abaixo, escreva uma **função** que divida a lista em duas novas listas.



3. Faça um **programa** que dado dois vetores, abaixo especificados:
  - a. Crie uma lista simplesmente encadeada, com elementos alternados dos vetores;
  - b. Exiba o maior e o menor valor, mostrando também seu endereço de memória.Vetores: `int vet_par[ ] = {2,4,6,8,10}`, `vet_impar[ ] = {1,3,5,7,9}`;
4. Faça uma **função** que remove um elemento inteiro de uma lista duplamente encadeada, obedecendo o protótipo:  
`int remove (int num);`  
Defina também a estrutura de dados struct No
5. Faça uma **função** que inverta lista simplesmente encadeada de números reais. Defina também a estrutura de dados que você irá usar.
6. Faça um programa que BUSQUE em uma LISTA DUPLAMENTE CIRCULAR da maneira mais eficiente possível. Use o esqueleto abaixo como referência.

```
struct No{
    int n;
    struct No *prox;
    struct No *ant;
}*inicio=NULL, *fim = NULL;
```

7. Faça um programa que controle uma FILA. Escolha qual a estrutura que usará e justifique sua escolha
8. Faça um programa que controle uma PILHA. Escolha qual a estrutura que usará e justifique sua escolha
9. Faça um programa que INCLUA em uma LISTA SIMPLEMENTE CIRCULAR de forma ordenada
10. Faça um função recursiva qualquer de livre escolha e explique sua escolha.