

1. Batman acaba de receber a mensagem abaixo, claramente enviada pelo Coringa. Conversando com o sr. Lucius Fox, seu fiel funcionário responsável pelo setor de tecnologia da Wayne Enterprise, Batman descobre que cada um dos números (de 2 dígitos) é facilmente traduzido para uma letra através da Tabela ASCII. Entretanto, a forma como a mensagem foi criada impede que o leitor saiba exatamente onde se inicia a mensagem.



a. Tendo aprendido programação em C ainda quando criança, Batman (ou Bruce Wayne) resolve criar uma **lista circular**. Cada número será inserido como chave de um nó desta lista, e iniciando a impressão destas chaves em todos nós possíveis, ao final imprimem-se todas as possíveis mensagens.

b. O sr. Lucius gostou da solução, mas se deu conta que Bruce está testando apenas a possibilidade dos números estarem dispostos em sentido **horário**. Então, ao mesmo tempo em que o Batman faz sua implementação, Lucius resolve criar uma **lista circular** a fim de imprimir todas possíveis mensagens no sentido **anti-horário**.

Quem conseguiu imprimir a mensagem do Coringa? **Escreva códigos** que sejam capazes de executar exatamente o mesmo que os códigos de Batman e Lucius. Dica: assumindo que `p` é um ponteiro para um determinado nó da lista circular ou da fila, você pode imprimir o número já como uma letra através `printf("%c", p->chave)`.

2. Divida uma lista encadeada circular em duas da seguinte forma: os nós com chave par devem permanecer na lista original, enquanto os nós com chave ímpar devem ser inseridos em uma nova lista circular.

3. Um grupo de sobreviventes do apocalipse zumbi está circundado por uma horda de zumbis. Não existe esperança de vitória contra essa horda e os sobreviventes planejam uma fuga a cavalo. A questão principal nesse momento é que o grupo só dispõe de um único cavalo. É necessário estabelecer um critério de sorteio para ver qual sobrevivente fará uso desse cavalo para escapar do massacre. Regras do sorteio:

- É sorteado um número `N` e o nome de um sobrevivente;
- Iniciando no sobrevivente eles começam a contar no sentido horário;
- O sobrevivente no qual a contagem `N` é finalizada é retirado do círculo;
- Todo sobrevivente que sair do círculo não entra mais no processo;
- O último sobrevivente é o feliz para escapar com o cavalo.

Implemente o sorteio utilizando uma lista encadeada circular.

4. Crie uma estrutura de dados para armazenar os N pontos (x,y) de um polígono. Implemente funções para:

- Inicializar a estrutura;
- Inserir pontos;
- Calcular o perímetro do polígono.

5. Copie as linhas abaixo para um arquivo “entrada.txt”:

```
1 1 2 aaa
1 1 5 bbb
1 2 3 ccc
1 4 9 ddd
1 4 8 eee
3 99 99 x
2 3 88 x
3 99 99 x
```

A seguir, crie um programa que leia esse arquivo utilizando no prompt “programa.exe < entrada.txt”. Considerando que cada linha tem 3 números e 1 string no formato “A B C S”, seu programa deve:

- Sempre que A = 1, insira um elemento em uma lista dinâmica duplamente encadeada (LDDE) com chave=C e info=S na posição B;
- Sempre que A = 2, remova o elemento da LDDE com chave = B (ignore C e S, neste caso);
- Sempre que A = 3, imprima todos nós da LDDE (ignore B, C e S, neste caso).

Acrescente outras linhas em “entrada.txt” para testes, caso julgue necessário.

6. Remova nós duplicados de uma lista duplamente encadeada. Evite uso de memória extra.

7. Implemente uma busca por chave para uma lista duplamente encadeada sabendo que seus elementos estão ordenados por chave. Se a chave buscada for mais parecida com a chave do 1º nó, faça uma busca sentido início->fim; caso a chave buscada seja mais parecida com a chave do último nó, faça uma busca no sentido fim->início.

8. Resolva o seguinte exercício: <http://br.spoj.com/problems/VIVO/>