

Exercício 4 – Lista duplamente encadeada e Circular

- 1) Implemente em um TAD uma operação com lista dupla encadeada que crie uma lista nova, com os mesmos elementos e na ordem invertida da lista passada por parâmetro:

```
Lista* cria_invertida(Lista* l)
```

- 2) Implemente um TAD de uma lista duplamente encadeada e implemente a função intersecção com a seguinte assinatura:

```
Lista* interseccao(Lista* l1, Lista* l2)
```

A funcao deve retornar uma lista com os elementos comuns a lista1 e lista2.

- 3) Implemente um TAD de um lista duplamente encadeada e implemente a função uniao com a seguinte assinatura:

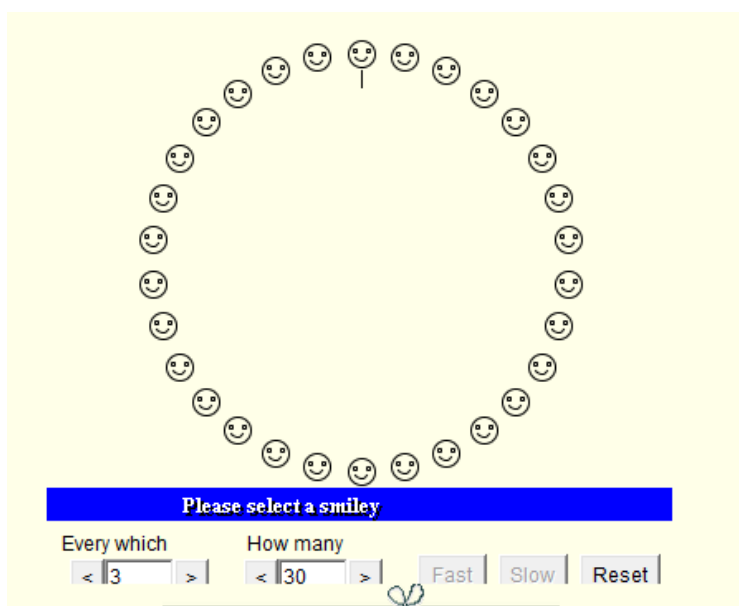
```
Lista* uniao(Lista* l1, Lista* l2)
```

A funcao deve retornar uma lista com todos os elementos da lista1 e lista2. Elimine os repetidos.

4) O problema de Josephus

Há um grupo de soldados cercados por uma força inimiga esmagadora. Não há esperança de vitória sem a chegada de reforços, mas existe apenas um cavalo disponível para escapar. Os soldados entram num acordo para determinar qual deles deverá escapar e pedir ajuda.

Um número n e o nome de um dos soldados é sorteado. Começando no soldado sorteado, começam a contar em sentido horário até atingir n. Este soldado é retirado do círculo. A contagem recomeça no soldado seguinte ao retirado. O soldado que restar deve montar no cavalo e ir buscar reforços. Quem será este soldado? Imprima o número e nome do soldado. Veja mais informações sobre este problema: <http://www.cut-the-knot.org/recurrence/flavius.shtml>



Solucione este problema usando TAD e lista circular.