

# Exercícios de Listas

1. Implemente as seguintes funções referentes a lista não ordenadas e simplesmente encadeada:

- a) `EliminaEnesimoElemento(Lista *L, int posicaoElemento);`
- b) `CopiaLista(Lista *original, Lista *copia);`
- c) Retornar a soma de inteiros de uma lista;
- d) Formar uma lista contendo a união dos elementos de duas listas.
- e) `search(Lista *L, int valorProcurado)` retorna um ponteiro do nó contendo `valorProcurado`, caso exista, ou um ponteiro `NULL`.

2. O problema de Josephus

Examinemos um problema que pode ser solucionado de maneira simples com uma lista circular. Ele é conhecido como problema de Josephus e postula um grupo de soldados circundados por uma força inimiga esmagadora. Não há esperança de vitória e existe apenas um cavalo disponível para a fuga. Os soldados entram em um acordo para determinar qual deles irá utilizar o cavalo para buscar ajuda. Eles formam um círculo e um valor  $n$  é sorteado. Um de seus nomes é sorteado também. Começando pelo soldado cujo nome foi sorteado, eles começam a contar ao longo do círculo em sentido horário. Quando a contagem alcança  $n$ , esse soldado é retirado do círculo, e a contagem reinicia com o soldado seguinte. O processo continua de maneira que, toda vez que  $n$  é alcançado, outro soldado sai do círculo. Todo soldado retirado do círculo não entra mais na contagem. O último soldado que sobrar deverá montar no cavalo e ir buscar ajuda.

O problema a ser resolvido é determinar a sequência na qual os soldados são eliminados do círculo e o soldado que escapará.

Por exemplo, suponha que  $n=3$  e que existem cinco soldados chamados: **A, B, C, D e E**. Imagine que o soldado sorteado seja o **A**. Contamos três soldados a partir de **A** para que **C** seja eliminado primeiro. Em seguida, começamos em **D** e contamos **D, E** e novamente **A** para que **A** seja eliminado. Depois contamos **B, D e E** e, finalmente, **B, D e B**, de modo que **D** seja o soldado a escapar.