



Prova 1 – Recursividade e Listas Encadeadas

Instruções:

1. **Provas e/ou questões idênticas (ou com indícios de cópia) terão a nota zerada.**
2. Ao término da prova, compacte a implementação de cada questão em um único arquivo (*com o seu nome completo*) e submeta no moodle na atividade denominada **“Prova 1 – Recursividade e Listas Encadeadas”**.

Questão 1 (2,0 pontos) – Escreva uma função recursiva MOD para calcular o resto da divisão de dois números inteiros positivos (x por y).

int MOD (int x, int y)

Por exemplo, o resultado de MOD(5,3) será 2.

Questão 2 (3,0 pontos) – Escreva a função **split** (na biblioteca “Lista.h”) a qual divide uma lista encadeada (recebida como parâmetro) em duas listas, conforme o critério dado pela função de predicado passada também como parâmetro.

A assinatura da função de predicado é dada a seguir:

typedef int (*FuncaoPredicado)(void *)

A função de predicado recebe um parâmetro do tipo “**void ***” e devolve um valor inteiro diferente de zero se o parâmetro tem o predicado desejado, caso contrário, retorna o valor zero.

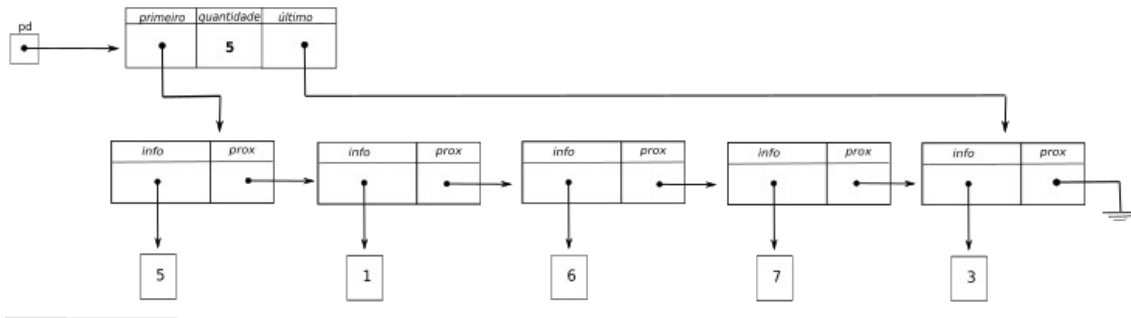
A assinatura da função **split** a ser implementada é dada a seguir:

pDLista split (pDLista lista, FuncaoPredicado fp)

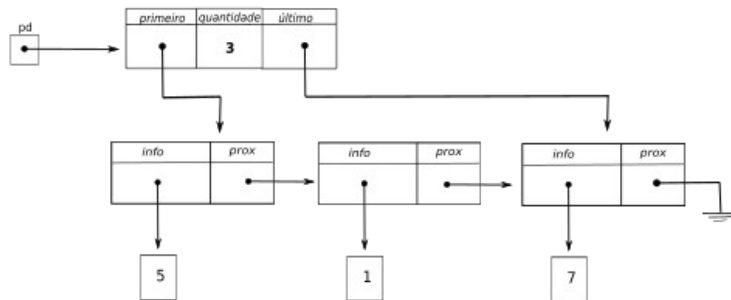
Exemplo: Considerando a lista encadeada a seguir, se a função de predicado dada verifica se o número inteiro é divisível por 3, então a função **split** gera como resultado uma nova lista contendo apenas os números divisíveis por 3 (ou seja, somente aqueles que satisfazem o predicado dado).

A figura a seguir ilustra o resultado produzido pela função **split**.

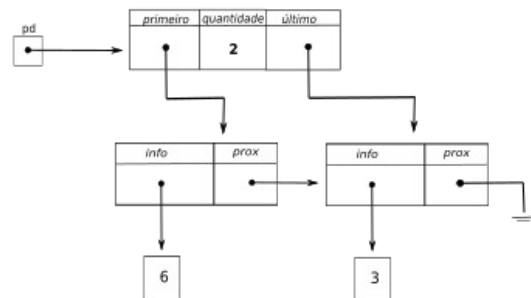
Lista Original



Lista 1 - lista original após a divisão

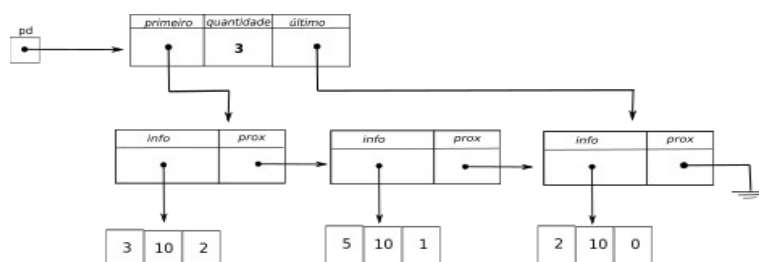


Lista 2 - lista com os elementos que satisfazem o predicado



Questão 3 (3,0 pontos) – Números inteiros podem ser decompostos e representados por meio de listas encadeadas, cujos nós são estruturas com 3 campos: **coeficiente**, **base** e **expoente**.

Por exemplo, a decomposição do número inteiro **352** ($3 \times 10^2 + 5 \times 10^1 + 2 \times 10^0$) pode ser representada pela lista encadeada a seguir.



Usando esta proposta de representação, escreva uma função que receba um número inteiro e produza como resultado uma lista encadeada representando a decomposição do número inteiro dado na base 10 (conforme ilustrado na figura acima).

Questão 4 (2,0 pontos) – Escreva uma função **RECURSIVA** que encontra o maior elemento de uma lista encadeada.

void* maiorElemento (**pDLista** lista, **void*** maiorAtual, *FuncaoComparacao* fc)