

MC202 — Estruturas de Dados

Professor

Ricardo da Silva Torres

<rtorres@ic.unicamp.br>

Primeira Lista de Exercícios

1. Considere a seguinte declaração para os nós de uma lista ligada:

```
typedef struct refnode {
    int valor;
    struct refnode *prox;
} tipoelemento, *tipolista, *ap_no;
```

- (a) Escreva uma função que remove o k -ésimo nó de uma lista ligada.
- (b) Escreva uma função que remove um elemento com valor v de uma lista ligada ordenada. Caso v não pertença à lista, uma mensagem de erro deve ser exibida.
- (c) Escreva uma função que recebe duas listas ligadas ordenadas $x = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_m)$ como parâmetros e retorna uma lista formada pelos elementos de x e y intercalados.
- (d) Escreva uma função que concatena duas listas encadeadas.
- (e) Escreva uma função que copie uma lista encadeada para um vetor.
- (f) Escreva uma função que verifica se duas listas ligadas dadas são iguais.
- (g) Escreva uma função não recursiva que retorna um apontador para a lista p invertida. Nenhum nó adicional deve ser criado.

```
ap_no inverte(ap_no p);
```

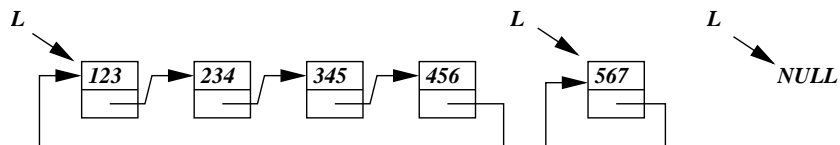
- (h) Escreva uma função que remove um elemento com valor v de uma lista ligada ordenada. Caso v não pertença à lista, uma mensagem de erro deve ser exibida.

```
void remove(ap_no* p, int v);
```

Por que um apontador para `ap_no` é passado como parâmetro? Se a lista tivesse um nó cabeça, poderíamos passar apenas um `ap_no`?

- (i) Faça uma rotina **recursiva** com dois parâmetros. No primeiro parâmetro deve ser enviado uma lista definida com o tipo acima. O segundo parâmetro receberá uma cópia desta lista contendo apenas os elementos de valor par.

2. Uma lista ligada circular é parecida com uma lista ligada simples. Ela também é identificada por uma variável apontando para o primeiro elemento, mas a diferença é que o último elemento, em vez de apontar para *NULL*, aponta para o primeiro elemento. Use a definição de mesmo tipo/estrutura apresentada na questão 3. anterior. Obs.: Se a lista está vazia, a variável aponta para *NULL*. Se a lista só tem um elemento, o campo *prox* aponta para o próprio elemento. Segue uma figura exemplificando a representação da lista circular.



- (a) Faça uma rotina para inserir um novo elemento no início da lista ligada circular.
(b) Faça uma rotina para remover o último elemento da lista ligada.