Lista 01 - Lista Linear Simplesmente Encadeada

1) O que é um apontador/ponteiro?

2) Qual constante do tipo apontador indica que o dado não aponta para nenhum outro dado válido?

3) Qual a diferença entre memória estática x memória dinâmica?

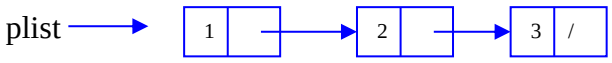
4) Conceitue Lista Linear.

5) Cite algumas operações feitas sobre Listas Lineares.

6) Quais são as duas formas usadas para representar Listas Lineares?

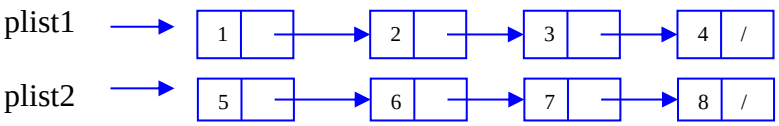
7) Desenhe uma lista linear simplesmente encadeada com 5 nós.

8) Suponha que temos uma lista encadeada conforme é exibido a seguir. Qual problema o uso da seguinte instrução pode causar?
Dado: plist = plist.getProx();

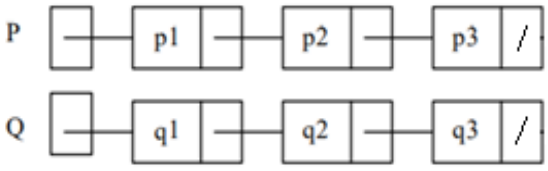


9) Suponha que exista uma lista encadeada conforme é mostrado a seguir. O que acontece se executarmos a seguinte instrução para estas duas listas?

plist1 = plist2;

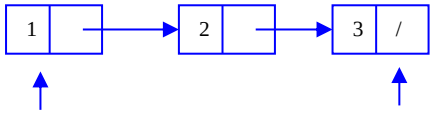


10) Partindo da situação mostrada no desenho abaixo, e sendo P e Q duas listas encadeadas dinâmicas, explique (com desenhos) o que acontece nas atribuições seguintes:
(obs: cada item parte da situação inicial mostrada abaixo)



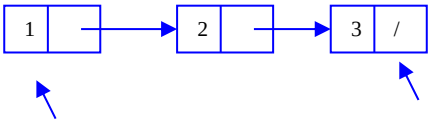
a) P.setProx(Q.getProx());
b) P = Q;
c) P = NULL;
d) P = P.getProx();
e) P.setInfo(P.getProx().getInfo());

11) Escreva o enunciado das seguintes operações:
a) _____



```
No aux = new No (elem);
ultimo.setProx(aux);
ultimo = aux;
```

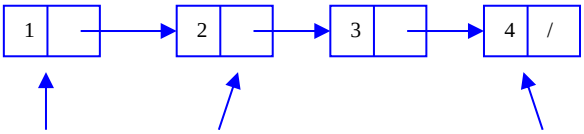
b) _____



primeiro ultimo

```
No aux = new No (elem);
aux.setProx(primeiro.getProx());
primeiro.setProx(aux);
```

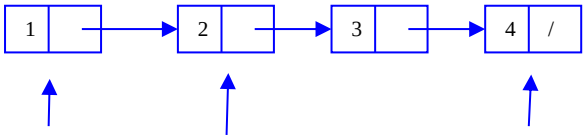
c) _____



primeiro atual ultimo

```
atual = atual.getProx();
primeiro.setProx(atual);
```

d) _____

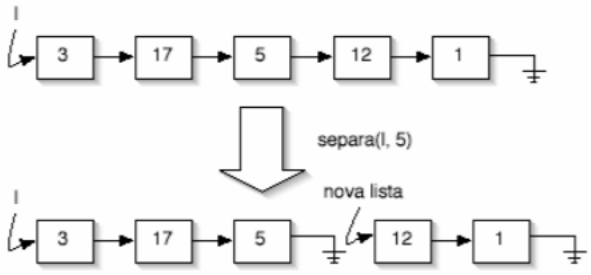


primeiro atual ultimo

```
ultimo = atual.getProx();
ultimo.setProx(null);
```

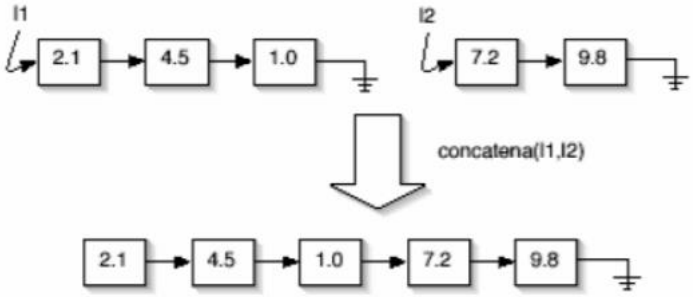
12) Seja a **Lista Simplesmente Encadeada** L, escreva um método para calcular a média dos valores da lista.

13) Considerando listas de valores inteiros, implemente uma função que receba como parâmetro um valor inteiro **n** e divida a lista em duas, de tal forma que a segunda lista comece no primeiro nó logo após a primeira ocorrência de **n** na lista original. A figura a seguir ilustra essa separação:



A função deve retornar um ponteiro para a segunda sub-divisão da lista original, enquanto l deve continuar apontando para o primeiro elemento da primeira subdivisão da lista.

14) Implemente uma função que, dadas duas listas encadeadas l1 e l2, concatene a lista l2 no final da lista l1, conforme ilustra a figura abaixo:



A função deve retornar a lista resultante da concatenação.

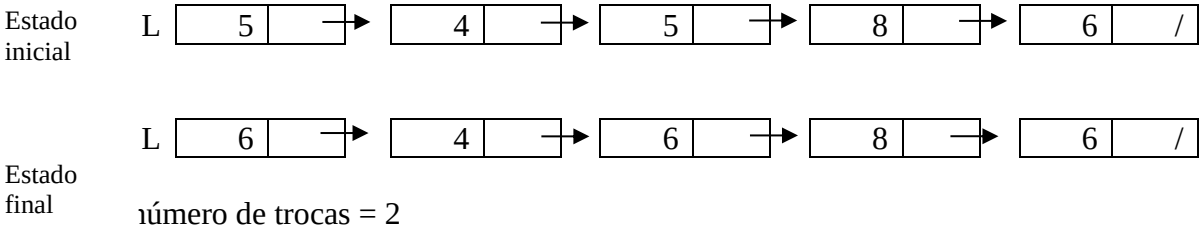
15) Considere uma lista simplesmente encadeada que armazena os seguintes dados de alunos de uma disciplina:

- Número de matricula
- Nome
- Média na disciplina

Implemente uma função que insira, em ordem crescente de número de matricula, os dados de um novo aluno na lista.
(OBS.: Crie novas classes Item, No e ListaSimples para atender a especificação do exercício)

16) Implemente uma função que utiliza duas **Listas Lineares Simplesmente Encadeadas** de inteiros, L1 e L2. Esta função deverá retornar VERDADEIRO se as listas forem idênticas (os mesmos elementos na mesma ordem), ou FALSO se pelo menos um elemento for diferente da ordem. Admita que as listas não estão vazias.

17) Dada uma **Lista Linear Simplesmente Encadeada**, implemente um método que receba como parâmetro dois valores (inteiros), e retorne o número de trocas efetuadas, resultante da troca de todas as ocorrências do primeiro valor(n1) pelo segundo(n2) dentro da lista. A lista não está vazia. Não precisa mudar os ponteiros.
Exemplo:
Parâmetros: n1=5 e n2=6



número de trocas = 2

18) Escreva um método que recebe como parâmetro um valor inteiro. Procure este valor na lista e se o valor não existir, ele deve ser inserido no final da lista; caso contrário, retorne a quantidade de nós que o valor aparece na lista. A lista é **Linear Simplesmente Encadeada com descritor**, e o nó da lista possui um campo do tipo inteiro. Não se sabe o estado da lista.

19) Implemente uma função que utiliza duas **Listas Lineares Simplesmente Encadeadas** de inteiros, L1(objeto do método) e L2(passada por parâmetro). Esta função deverá retornar TRUE se uma lista L1 está contida na lista L2 (independente da ordem), ou FALSE se pelo menos um elemento da lista L1 não foi encontrado na L2. Você não sabe como estão as listas (vazias ou não vazias, se uma tem mais ou menos elementos que a outra).

20) Dada a **Lista Linear Simplesmente Encadeada**, implemente um método que receba como parâmetro um número de um partido. O método deverá remover o(s) elemento(os) que possui(rem) a chave igual ao número do partido passado como parâmetro e retornar o número de elementos removidos. A Lista pode não ter nenhum nó com chave igual ao partido, ou pode ter um ou vários nós com chave igual ao partido. Sabe-se que a lista não está vazia.

21) Assuma que uma **Lista Linear Simplesmente Encadeada** é construída de elementos com dois campos, *info* e *prox*, sendo que o campo *prox* de cada elemento é usado para apontar para o próximo na lista. Qual das seguintes é uma implementação correta em Java da operação *insira p depois de q*, onde *q* aponta para um elemento da lista e *p* para um elemento a ser inserido?

(A)
```
q.setProx(p.getProx());
p.setProx(q);
```
(B)
```
q.setProx(p.getProx());
p.setProx(q.getProx());
```
(C)
```
p.setProx(q.getProx());
q.setProx( p);
```
(D)
```
p.setProx(q);
q.setProx(p.getProx());
```
(E)
```
q.setProx(p);
p.setProx(q.getProx());
```

22) (FCC - 2009 - TJ-PI - Analista Judiciário - Análise de Sistemas - Desenvolvimento) Uma lista ligada é uma estrutura que corresponde a uma sequência lógica de entradas ou nós. Cada nó armazena a localização do próximo elemento na sequência, ou seja, de seu nó sucessor. Nessa estrutura,

(A) para estabelecer a ligação entre um nó já pertencente a uma lista e um novo nó, basta fazer com que o novo nó referencie no, campo *next*, o nó que anteriormente era referenciado pelo nó original, desde que esse campo não tenha o valor nulo.

(B) a existência de um ponteiro apontando para o 1º elemento e outro para o fim da lista permite que a inserção ou deleção de dados de um nó que esteja no meio da lista seja rapidamente executada.

(C) enquanto a entrada que determina o topo da lista é mantida em um nó descritor dessa lista, a entrada que marca o fim da lista é mantida fora do descritor.

(D) o armazenamento de uma lista requer uma área contígua de memória para permitir a otimização no processamento de criação e remoção de nós da lista.

(E) o armazenamento de uma lista não requer uma área contígua de memória. Como listas são estruturas dinâmicas, normalmente são definidos procedimentos que permitem criar e remover nós na memória.