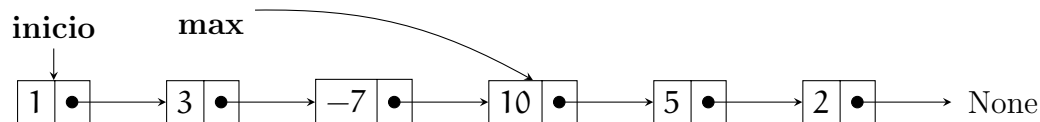


Estruturas de Dados**Prática: listas encadeadas**

1. (5 pontos) Considere uma estrutura de lista encadeada. Queremos implementar de forma eficiente (rápida) a busca pelo elemento de maior valor na lista. Por exemplo, uma operação `busca_max()` sobre a lista contendo os números $\{1, 3, -7, 10, 5, 2\}$ deve fornecer como resposta o valor 10. Queremos executar essa operação sem percorrer toda a lista. Para isso, devemos manter uma referência `max` para o maior elemento da lista. Desse modo, `busca_max()` simplesmente retorna o item apontado por `max`.



Que impactos esse novo atributo da lista terá sobre as operações de inserção no início da lista e de remoção de um elemento arbitrário? Certamente o custo das operações de inserção e remoção na lista serão afetados. No exemplo acima, o que ocorre se 10 é removido da lista? O que ocorre se inserimos 16?

Apresente novas versões das operações de inserção e remoção na lista de modo a manter o atributo `max` sempre atualizado e coerente.

2. (5 pontos) Polinômios podem ser representados eficientemente por meio de listas encadeadas. Neste exercício, você escreverá um tipo abstrato de dados para representar um polinômio usando lista encadeada.

A título de exemplo o polinômio $3x^6 - x + 10$ pode ser representado pela lista encadeada contendo os dados de cada termo (coeficiente e expoente) em um nó, como mostrado no diagrama a seguir.



Implemente pelo menos as seguintes operações sobre polinômios.

1. **calcular.** Computa o valor do polinômio quando x assume um determinado valor. Por exemplo, quando $x = 2$ o valor de $x^2 - 1$ é 3.
2. **somar.** Calcula a soma de dois polinômios. Note que a soma é um novo polinômio. Assim ao somar $x^2 - 1$ com $x^3 + 2x^2$ obtemos $x^3 + 3x^2 - 1$.
3. **derivada.** Calcula a derivada de um polinômio, gerando um novo. Por exemplo, a derivada de $x^3 + 2x$ é $3x - 2$.

Evidentemente, você precisa implementar criação de polinômios vazios e inserção de termos (com expoente e coeficiente) em um polinômio.

Código de apoio

```
Class Poli_No:
    def __init__(self, coef, exp):
        self.coef = coef
        self.exp = exp

Class Poli:
    # cria polinomio vazio
    def __init__(self):
        self.inicio = None

    def inserir_termo(self, coef, exp):
        puts("ainda nao implementado")

    def calcular(self, x):
        puts("ainda nao implementado")

    def somar(self, poli):
        soma = Poli()
        puts("ainda nao implementado")
        return soma

    def derivada(self):
        deriv = Poli()
        puts("ainda n o implementado")
        return deriv
```