

## Ordenação de Pontos

Python é uma linguagem fortemente orientada a objetos. Isso inclui a forma ela lida com expressões e operadores. Por exemplo, quando escrevemos a expressão `x < y`, o que o Python faz é enviar a mensagem `__lt__(y)` ao objeto `x`, indicando que ele deve comparar-se com o objeto `y`. A mensagem `__lt__` é usada para sobrecarregar o operador `<`, assim como a mensagem `__add__` sobrecarrega o operador `+` e a mensagem `__abs__` sobrecarrega a operação `abs()`.

Neste exercício, você deverá criar uma classe para guardar pontos no plano cartesiano. Além de criar o construtor, você deverá sobrecarregar o método `__lt__` de modo que seja possível ordenar os pontos de acordo com os seguintes critérios:

1. O ponto mais próximo da origem deve aparecer primeiro. Por exemplo, `(2,2)(2,2)` está mais perto da origem do que `(0,3)(0,3)`;
2. Se dois pontos estiverem à mesma distância da origem, o ponto que estiver mais à esquerda vem primeiro. Exemplo: `(0,5)(0,5)` e `(3,4)(3,4)`;
3. Se os dois pontos estiver à mesma distância da origem e o mesmo valor da coordenada `xx`, então o ponto que estiver mais abaixo vem primeiro. Exemplo: `(6,-8)(6,-8)` e `(6,8)(6,8)`.

O nome da sua classe deverá ser `ponto` e ela deverá conter os seguintes métodos:

- `__init__(self, x, y)`: o construtor deverá receber as coordenadas do ponto e guardar os atributos `self.x` e `self.y`;
- `__lt__(self, outro)`: este método sobrecarregará o operador `<`. Ele deverá receber como argumento um outro objeto da classe `ponto` e retornar `True` se e somente se o ponto representado por `self` deve aparecer antes do que o ponto representado por `outro`.

Em seguida, faça um programa principal que lê uma série de pontos e os guarda em uma lista para que seja possível ordená-los. Não implemente a função de ordenação. Em vez disso, use o método `list.sort()` para ordenar essa coleção de pontos. O Python será capaz de ordenar os pontos contanto que o operador `<` esteja sobrecarregado.

Ao final, imprima a lista ordenada de acordo com critérios apresentados acima.

## Entrada e Saída

A entrada será uma lista contendo  $NN$  tuplas, isto é,  $[(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)]$ . Cada tupla  $(x_i,y_i)$  representa as coordenadas de um ponto no plano cartesiano. Imprima a coleção ordenada usando o formato ilustrado nos casos de exemplo.

**Exemplos de Entrada e Saída**

Entrada	<code>[(0, 1), (3, 4), (-6, 8)]</code>
Saída	<code>0 1 3 4 -6 8</code>
Entrada	<code>[(-1, 0), (-1, 68), (0, 1), (15, -55), (15, 20)]</code>
Saída	<code>-1 0 0 1 15 20 15 -55 -1 68</code>