



# Quadrática

## 1) Parte 1: Avaliação Quadrática

### Descrição

Neste problema, você escreverá um programa curto para calcular o resultado da avaliação de uma expressão  $ax^2 + bx + c$  em um valor particular de  $x$ .

As primeiras quatro linhas de seu programa devem configurar variáveis para conter os valores relevantes ( $a, b, c, x$ ). Portanto, as primeiras quatro linhas do seu programa devem ser semelhantes a, por exemplo:

```
a = 1
b = 2
c = 3
x = 0.5
```

Seu programa deve vincular a variável `out` ao valor associado à avaliação do quadrático especificado por `a`, `b` e `c` com o valor de `x` fornecido.

### Notas

- Python não é exigente com nomes de variáveis, mas nosso verificador é. É importante que você chame as variáveis mencionadas acima de `a`, `b`, `c`, `x` e `out`, em vez de dar-lhes outros nomes.

### Submissão

Quando estiver pronto, faça upload do seu arquivo Python no **Problema 0.2.1** no Gradescope. Lembre de nomear seu arquivo `p0_2_1.py`.

## 2) Parte 2: Raízes Quadráticas

### Descrição

Agora você escreverá um programa para calcular as raízes da expressão  $ax^2 + bx + c$ .

As primeiras três linhas de seu programa devem configurar variáveis para conter os valores relevantes ( $a, b, c$ ). Portanto, as primeiras três linhas do seu programa devem ser semelhantes a, por exemplo:

```
a = 1
b = 2
c = 3
```

Seu programa deve vincular as raízes do polinômio à variável `out`. Se houver apenas uma raiz (no caso  $a = 0$ ), simplesmente ligue-a a `out`. Caso haja duas raízes  $x_1$  e  $x_2$ , conecte-as a `out` com uma vírgula separando os dois valores, como `out = x1, x2` por exemplo.

### Notas

- Tente simular seu código manualmente para alguns valores diferentes da variável acima para ter certeza de que tem um programa funcionando. Envie-o apenas depois de testá-lo, manualmente e em

seu próprio computador usando Python.

- Python não é exigente com nomes de variáveis, mas nosso verificador é. É importante que você chame as variáveis mencionadas acima de `a`, `b`, `c` e `out`, em vez de dar-lhes outros nomes.
- Python é realmente incrível; tem suporte integrado para números complexos! Tente, por exemplo, avaliar o seguinte: `print(1 + (-2)**0.5)`. Você verá o Python imprimir sua representação integrada de um número complexo. Observe também que Python usa uma notação comum em engenharia: a unidade imaginária  $\sqrt{-1}$  é representada por  $j$  ao invés de  $i$ .

## Submissão

Quando estiver pronto, faça upload do seu arquivo Python no **Problema 0.2.2** no Gradescope. Lembre de nomear seu arquivo `p0_2_2.py`.