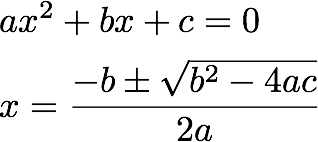
Consider the quadratic equation and its canonical solution:



The part b^2-4\*a\*c is called the discriminant. Suppose we want to provide an API with two different strategies for calculating the discriminant:

1. In OrdinaryDiscriminantStrategy , If the discriminant is negative, we return it as-is. This is OK, since our main API returns Complex  numbers anyway.
2. In RealDiscriminantStrategy , if the discriminant is negative, the return value is NaN (not a number). NaN propagates throughout the calculation, so the equation solver gives two NaN values. In Python, you make such a number with float('nan').

Please implement both of these strategies as well as the equation solver itself. With regards to plus-minus in the formula, please return the + result as the first element and - as the second. Note that the solve() method is expected to return complex values.