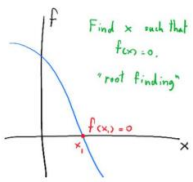
P10. (difficulty 4*) Root finding is to solve an equation f(x) = 0 for x. One of the most



intuitive algorithm is Bisection algorithm. Bisection starts out with an interval known to contain the root x* , where f(x*) =0. An interval is defined by lower limit a and upper limit b. It works by narrowing the interval down to a very small size. Then, the root x* ≈ (a + b)/2.

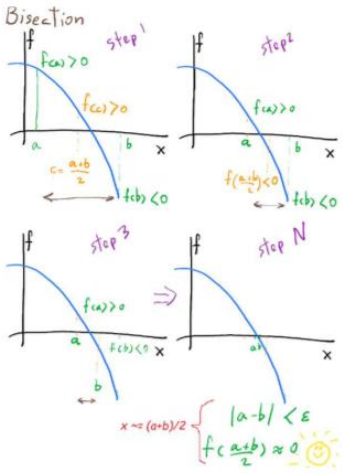
Bisection pseudo-code:

(adapted from https://en.wikipedia.org/wiki/Bisection_method)

```
INPUT:       Function f, endpoint values a, b, tolerance TOL

CONDITIONS:  a < b, either f(a) < 0 and f(b) > 0
                    or     f(a) > 0 and f(b) < 0
OUTPUT: value which differs from a root of f(x)=0 by less than TOL

While | a - b | > TOL
       c = (a + b)/2 # new midpoint
       if f(c) = 0 then Return c
       If sign(f(c)) = sign(f(a)) then
              a = c
       else
              b = c
EndWhile
Return (a+b)/2
```



Write a function implementing Bisection pseudo-code above. Use the P10 template. (QBisectionTemplate.py. The template is only to ensure the exact display format and allows smooth auto-grading. **Caution:** without proper edit to the template, the loop will stuck forever.)

Hint: abs(-4) gives 4 and abs(5) gives 5.

Example:

==========

a:-5

b:0

TOL:0.0001

f(-2.791)=-0.00016

==========

a:1

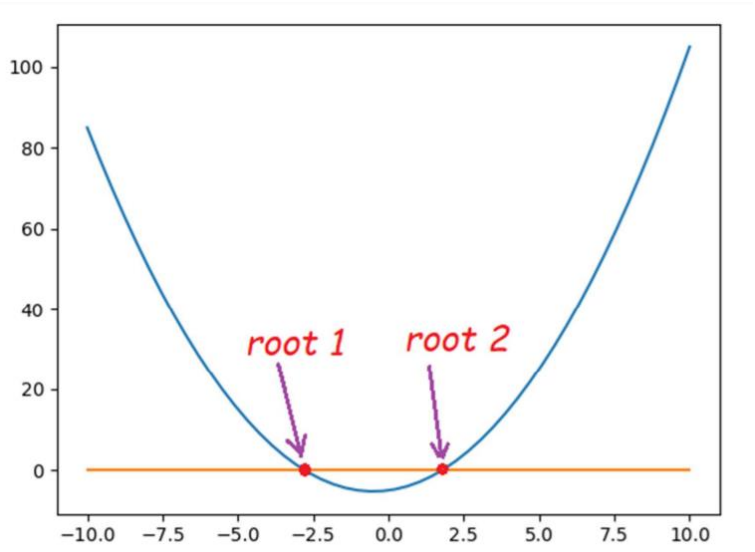
b:5

TOL:0.01

f(1.793)=0.00771

==========

Note: this example actually has 2 roots..



Here is P10_template.py

```python
"""
Write a function implementing Bisection.
"""

def fpolynomial(x, ws):
    n = len(ws)

    y = 0
    for i in range(n):
        y += ws[i] * x**i

    return y

# Don't edit above this line.
# ===========================================


def bisection(a, b, TOL):

    # f(a)
    fa = fpolynomial(a, [-5, 1, 1])

    while abs(a - b) > TOL:
        c = (a + b) / 2

        # f(c)
        fc = fpolynomial(c, [-5, 1, 1])

        if fc == 0:
            return c


        # You complete the algorithm here ...
        # ... something missing ...
        # Caution!!!
        # without proper code here, the loop will stuck forever!!!


    return (a + b)/2
```

```
## Don't edit below this line.
## ================================================

if __name__ == '__main__':

    xa = float(input('a:'))
    xb = float(input('b:'))
    tol = float(input('TOL:'))

    root = bisection(xa, xb, tol)
    print('f(', root,')=', fpolynomial(root, [-5, 1, 1]))
```