```python
# From Lecture 4, Understanding Root Finding

# root code

def findRoot1(x, power, epsilon):
    low = 0
    high = x
    ans = (high+low)/2.0
    while abs(ans**power - x) > epsilon:
        if ans**power < x:
            low = ans
        else:
            high = ans
        ans = (high+low)/2.0
    return ans

##print findRoot1(25.0, 2, .001)
##print findRoot1(27.0, 3, .001)
##print findRoot1(-27.0, 3, .001)


# so can't find cube root of negative number

def findRoot2(x, power, epsilon):
    if x < 0 and power%2 == 0:
        return None
    # can't find even powered root of negative number
    low = min(0, x)
    high = max(0, x)
    ans = (high+low)/2.0
    while abs(ans**power - x) > epsilon:
        if ans**power < x:
            low = ans
        else:
            high = ans
        ans = (high+low)/2.0
    return ans

##print findRoot2(25.0, 2, .001)
##print findRoot2(27.0, 3, .001)
##print findRoot2(-27.0, 3, .001)
```

```
##
##print findRoot2(0.25, 2, .001)
##print findRoot2(-0.125, 3, .001)

def findRoot3(x, power, epsilon):
    if x < 0 and power%2 == 0:
        return None
    # can't find even powered root of negative number
    low = min(-1.0, x)
    high = max(1.0, x)
    ans = (high+low)/2.0
    while abs(ans**power - x) > epsilon:
        if ans**power < x:
            low = ans
        else:
            high = ans
        ans = (high+low)/2.0
    return ans

print findRoot3(25.0, 2, .001)
print findRoot3(27.0, 3, .001)
print findRoot3(-27.0, 3, .001)

print findRoot3(0.25, 2, .001)
print findRoot3(-0.125, 3, .001)

def testFindRoot():
    epsilon = 0.0001
    for x in (0.25, -0.25, 2, -2, 8, -8):
        for power in range(1,4):
            print('Testing x = ' + str(x) +\
                  ' and power = ' + str(power))
            res = findRoot3(x, power, epsilon)
            if res == None:
                print('    No root')
            else:
                print('    ' + str(res**power) + ' ~= ' + str(x))
```