

4.2

코드

```
from bisection import*
def f(x): return x**3 - 10.0*x**2 + 5.0
x = bisection(f,0.0,1.0,tol=1.0e-4)
print('x=','{:6.4f}'.format(x))

import math
import error
from numpy import sign

def bisection(f,x1,x2,switch=0,tol=1.0e9):
    f1 = f(x1)
    if f1== 0.0: return x1
    f2 = f(x2)
    if f2 == 0.0: return x2
    if sign(f1)==sign(f2):
        error.err('Root is not bracketed')
    n = int(math.ceil(math.log(abs(x2-x1)/tol)/math.log(2.0)))

    for i in range(n):
        x3 = 0.5*(x1+x2); f3 = f(x3)
        if(switch==1) and (abs(f3)>abs(f1)) and abs(f3) > abs(f2):
            return None
        if sign(f2)!=sign(f3):x1 = x3; f1 = f3
        else: x2 = x3;f2=f3
    print("반복횟수 : ", n)
    return (x1+x2)/2.0
```

실행결과

반복횟수 : 14

x= 0.7346

Process finished with exit code 0

4.4

코드

```

from ridder import*

def f(x):
    a = x**3
    b = 10*(x**2)
    return a - b + 5
t = ridder(f,0.6,0.8)
print(t)
print("f(x)=", f(t))

import ...
def ridder(f,a,b,tol=1.0e-9):
    fa = f(a)
    if fa==0.0: return a
    fb = f(b)
    if fb == 0.0: return b
    for i in range(30):
        c = 0.5*(a+b);fc = f(c)
        s = math.sqrt(fc**2 - fa*fb)
        if s==0.0: return None
        dx = (c-a)*fc/s
        if(fa-fb) <0.0: dx = -dx
        x = c+dx; fx=f(x)
        if i>0:
            if abs(x-x0ld) < tol*max(abs(x),1.0):
                print("i:",i+1)
                return x
            x0ld = x
        if sign(fc) == sign(fx):
            if sign(fa)!=sign(fx): b=x; fb = fx
            else: a=x;fa=fx
        else:
            a=c;b=x;fa=fc;fb=fx
    return None
print('too many iterations')

```

실행 결과

반복: 4

0.7346035077893032

$f(x) = 8.881784197001252e-16$

Process finished with exit code 0

4.7

코드

```
def f(x): return x**4 - 6.4*x**3 + 6.45*x**2 + 20.538*x - 31.752
def df(x): return 4.0*x**3 - 19.2*x**2 + 12.9*x + 20.538

def newtonRaphson(x, tol=1.0e-9):
    for i in range(40):
        dx = -f(x)/df(x)
        x = x+dx
        if abs(dx) < tol: return x, i
    print(f'too many iteration')
    return 0, 0

root, numIter = newtonRaphson(2.0, 10**(-10))
print(f'root = {root}')
print(f'NUMBER OF ITERATION = {numIter}')
```

실행결과

```
root = 2.09999999786199406
NUMBER OF ITERATION = 22

Process finished with exit code 0
```