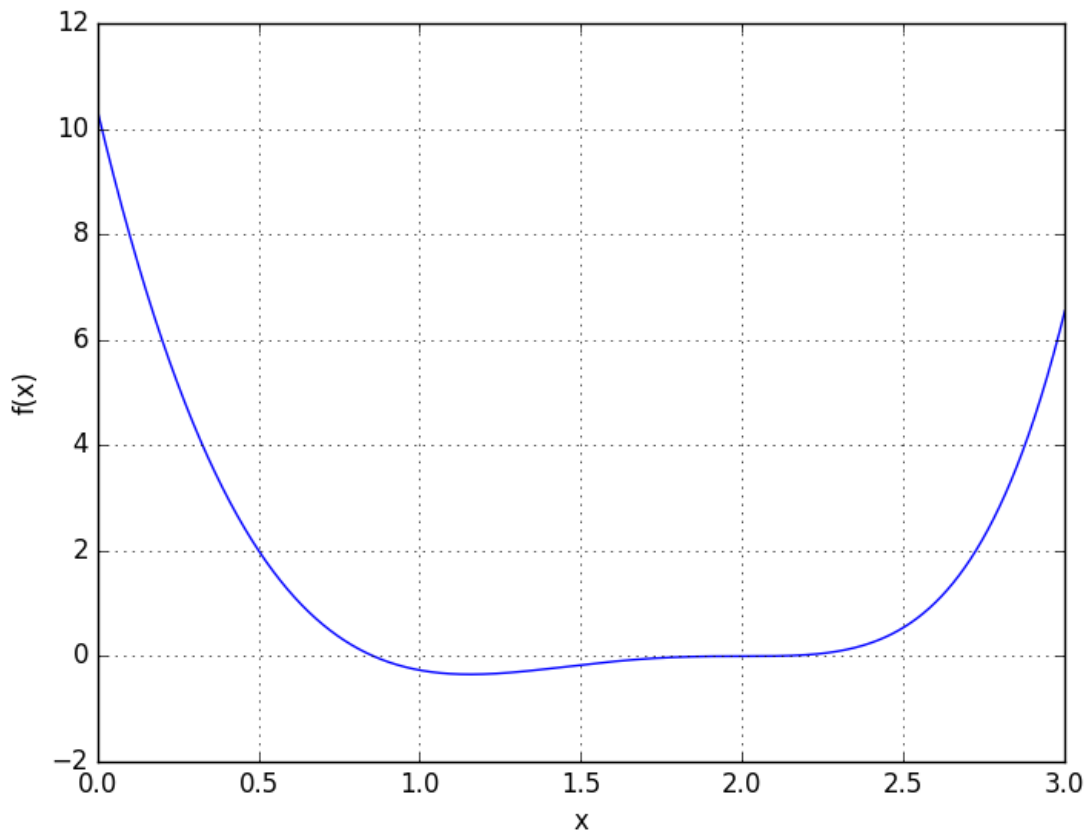


Στην αρχή έχω τα απαραίτητα imports για numpy, scipy και matplotlib:

```
1 from numpy import *
2 import numpy
3
4 from scipy import *
5 import scipy
6
7 import matplotlib.pyplot as plt
```

Η συνάρτηση ορίζεται και γίνεται το plot της:

```
9 t = arange(0,3.001,step=.001)
10
11 def f(t):
12     return (14*t)*(e**(t-2)) - 12*(e**(t-2)) - 7*(t**3) + 20*(t**2) - 26*t + 12
13
14 plt.xlabel("x")
15 plt.ylabel("f(x)")
16 plt.plot(t, f(t))
17 plt.grid(True)
18 plt.show()
```



Στο πρώτο κομμάτι της άσκησης η προσέγγιση της ρίζας γίνεται με την μέθοδο της διχοτόμησης:

```
20 def bisect(a, b):
21     root = (a + b) / 2
22
23     # number of times to repeat to achieve 6 points accuracy
24     N = int(ceil((log(b - a) - log(0.000005)) / log(2))) + 20
25
26     for i in range(0, N):
27         if (f(a) < 0 and f(root) > 0) or (f(a) > 0 and f(root) < 0):
28             b = root
29         elif (f(b) < 0 and f(root) > 0) or (f(b) > 0 and f(root) < 0):
30             a = root
31         root = (a + b) / 2
32
33     return ("f(%.6f) = %.6f" % (root, f(root)))
34
35 print("Root in [0,1.5]: %s" % (bisect(0,1.5)))
36 print("Root in [1.5,3]: %s" % (bisect(1.5,3)))
```

```
$ python ex1.py
$ f(0.952374) = 0.000000
```

Στην συνέχεια υπολογίζω τις ρίζες με την μέθοδο Newton-Raphson: