**Exercise 1**
SEE CODE

**Exercise 2**
SEE CODE

Exercise 3

- V1: x = 2.506184145588769, iterations = 27
- V2: x = 3.141592653589793, iterations = 6
- V3: x = error (division by zero), stopped at iteration = 4
- V4: x = error (division by zero), stopped at iteration = 96
- V5: x = error (division by zero), stopped at iteration = 1

| | fsolve | optimize.newton | optimize.fsolve |
|---|---|---|---|
| v1 | 2.506184 | 1.000200e+00 | 1.000000e+00 |
| v2 | 3.141593 | 3.141593e+00 | 3.141593e+00 |
| v3 | NA | 3.402824e+38 | 3.402824e+38 |
| v4 | NA | -3.834488e-02 | 1.759157e-03 |
| v5 | NA | 9.998667e-05 | 0.000000e+00 |

This table gives value of fsolve, optimize.newton and optimize.fsove for each example.

- For v1, optimized.fsolve outputs 1, which is wrong.
- For v2, 3 methods give same value
- For V3, V4, V5, optimize function output value although optimize.fsolve for v5 failed since it does not have a real root

Try poor initial guess in order to test fprime: SEE CODE

```
'''
to see the advantge for giving fprime
'''
# the same
fsolve(lambda x: x**4, -100)
fsolve(lambda x: x**4, -100,lambda x: 4*x**3)

#without fprime, cannot converge; with fprime, can converge within maxiter = 100
op.newton(lambda x: x**4, -100,maxiter=100)
op.newton(lambda x: x**4, -100, fprime = lambda x: 4*x**3,maxiter=100)

# with prime exactly zero and converge faster
op.fsolve(lambda x: x**4, -100)
op.fsolve(lambda x: x**4, -100, fprime = lambda x: 4*x**3)
```

- Solve function, with fprime and without fprime, output same values.
- Newton function, with max iteration = 100, cannot converge to a solution without fprime, can converge to a solution with fprime.
- Op.fsolve function gives exactly zero value and converges faster with fprime.

**Exercise 4**
SEE CODE

| | w fprime (val) | w/o fprime (val) | w fprime (iter) | w/o fprime (iter) |
|---|---|---|---|---|
| 0 | 1.396984e-08 | 1.396984e-08 | 32 | 32 |
| 10 | 3.162278e+00 | 3.162278e+00 | 9 | 9 |
| 49 | 7.000000e+00 | 7.000000e+00 | 8 | 8 |
| 100 | 1.000000e+01 | 1.000000e+01 | 7 | 7 |
| 200 | 1.414214e+01 | 1.414214e+01 | 7 | 7 |
| 400 | 2.000000e+01 | 2.000000e+01 | 6 | 6 |
| 800 | 2.828427e+01 | 2.828427e+01 | 6 | 6 |
| 1500 | 3.872983e+01 | 3.872983e+01 | 5 | 5 |

The table shows some example input and corresponding output from different function with fprime or without fprime. The number of iteration does not change by sqrt function with fprime or without fprime.