1 Let $f(x) = x^3 - e^x$.

a. Show that $f(x) = 0$ has a solution p on $[1.5, 2]$.

b. We wish to solve $f(x) = 0$ on $[1.5, 2]$ by the fixed-point iteration. Justify that fixed-point problems $x = g_i(x)$ with the following g_i, $i = 1, 2, 3$ are equivalent to the original root-finding problem $f(x) = 0$.

   i. $g_1(x) = x + (x^3 - e^x)$.

   ii. $g_2(x) = e^{x/3}$.

   iii. $g_3(x) = \sqrt{x^{-1}e^x}$.

c. For each g_i above, determine whether the fixed-point iteration $p_n = g_i(p_{n-1})$ with $p_0 = 1.75$ converges to p.

Solution a. Notice that $f(1.5) \approx -1 < 0$ and $f(2) \approx 0.6 > 0$, so by the intermediate value theorem, $\exists p \in [1.5, 2]$ such that $f(p) = 0$.

b. i. $g_1(x) = x \iff x = x + (x^3 - e^x) \iff x^3 - e^x = 0$, since subtracting by x is one-to-one.

   ii. $g_2(x) = x \iff e^{x/3} = x \iff (e^{x/3})^3 = x^3 \iff x^3 - e^x = 0$, since cubing and subtracting by x^3 are one-to-one.

   iii. $\sqrt{x}$ restricted to $x > 0$ is a bijection, so

$$g_3(x) = x \iff x = \sqrt{x^{-1}e^x} \iff x^2 = x^{-1}e^x \iff x^3 - e^x = 0.$$

   We can safely multiply both sides by x, since $x \geq 1.5 > 0$, and because subtraction by a particular number is one-to-one.

c. i. Notice that $g_1'(x) = 1 + 3x^2 - e^x$ and $g_2''(x) = 6x - e^x > 0$ on $[1.5, 2]$. Hence, $g_1'(x)$ is strictly increasing on the same interval, which means

$$g_1'(x) \geq g_1'(1.5) \geq 3.$$

   Thus,

$$|p_n - p_{n-1}| = |g_1(p_{n-1}) - g_1(p_{n-2})| = |g_1'(\xi_n)||p_{n-1} - p_{n-2}| \geq 5|p_{n-1} - p_{n-2}| \geq \cdots \geq 5^{n-1}|p_1 - p_0|.$$

   So, the sequence cannot be Cauchy, so it cannot converge.

   ii. Note that $g_2'(x) = (1/3)e^{x/3}$. Since $e^{x/3}$ is increasing,

$$|g_2'(x)| \leq \frac{e^{2/3}}{3} < \frac{e}{3} < 1.$$

   Thus, by the fixed-point theorem, the sequence generated with $p_0 = 1.75$ converges to the fixed point.

   iii. Once again,

$$|g_3'(x)| = \left|\frac{x-1}{2x}\sqrt{\frac{e^x}{x}}\right| \leq \frac{2-1}{2 \cdot 1.5}\sqrt{\frac{e^2}{1}} = \frac{e}{3} < 1,$$

   so again by the fixed-point theorem, the sequence generated converges.

2 Let $f(x) = x^2 - 4x + 3$. It is known that the two solutions of $f(x) = 0$ are 1 and 3.

a. Manually implement the Newton's method to solve $f(x) = 0$ with $p_0 = 4$. You may stop when $|p_N - p_{N-1}| < 10^{-3}$. You may use calculators or computers.

b. What is the relative error of your final p_n with respect to the exact solution $p = 3$?

c. Prove that in this case, for all $n \geq 1$,

$$0 < p_n - 3 \leq \frac{1}{2}(p_{n-1} - 3)^2.$$

Solution a. Newton's method generates a sequence via

$$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)} = p_n - \frac{(p_n - 1)(p_n - 3)}{2(p_n - 2)}.$$

This gives us the following table:

n	p_n
0	4
1	3.25
2	3.025
3	3.000 304 878 05
4	3.000 000 046 46

b. The relative error is

$$\frac{|p_4 - p|}{|p|} = \frac{3.00000004646 - 3}{3} \approx 1.5 \times 10^{-8}.$$

c. Notice that

$$\begin{aligned}
\frac{p_n - 3}{(p_{n-1} - 3)^2} &= \frac{1}{(p_{n-1} - 3)^2}\left(p_{n-1} - 3 - \frac{(p_{n-1} - 1)(p_{n-1} - 3)}{2(p_{n-1} - 2)}\right) \\
&= \frac{1}{p_{n-1} - 3} - \frac{p_{n-1} - 1}{2(p_{n-1} - 2)(p_{n-1} - 3)} \\
&= \frac{2p_{n-1} - 4 - p_{n-1} + 1}{2(p_{n-1} - 2)(p_{n-1} - 3)} \\
&= \frac{p_{n-1} - 3}{2(p_{n-1} - 2)(p_{n-1} - 3)} \\
&= \frac{1}{2(p_{n-1} - 2)}.
\end{aligned}$$

It now suffices to show that $p_{n-1} - 2 \geq 1$, which we'll prove by induction.

Base step:

$n = 0$, $p_0 - 2 = 2 \geq 1$, so the base step holds.

Inductive step:

Suppose that $p_{n-1} - 2 \geq 1$. We wish to show that $p_n - 2 \geq 1$.

Notice that we also get

$$p_{n-1} - 1 \geq 2 \quad \text{and} \quad p_{n-1} - 3 \geq 0.$$

Hence,

$$p_n - 2 = (p_{n-1} - 2) - \frac{(p_{n-1} - 1)(p_{n-1} - 3)}{2(p_{n-1} - 2)} \geq 1 - 0 = 1,$$

so the inductive step holds.

Thus, by induction,

$$p_n - 2 \geq 1 \implies \frac{1}{p_n - 2} \leq 1,$$

so

$$0 < \frac{p_n - 3}{(p_{n-1} - 3)^2} = \frac{1}{2(p_{n-1} - 2)} \leq \frac{1}{2} \implies 0 < p_n - 3 \leq \frac{1}{2}(p_{n-1} - 3)^2.$$

---

3 Let $f(x) = x^2 - 4x + 3$ as in the previous problem. Numerically implement the Newton's method to solve $f(x) = 0$ with $p_0 = 1.99$. You may stop the iteration when $|p_N - p_{N-1}| < 10^{-5}$.

Repeat this with new initial data $q_0 = 2.01$, which is close to p_0. What do you find?

Solution My code was written in Python.

```python
def newton(f, df, init, eps, N=100):
    """
    Implementation of Newton's method
    Parameters:
        f    - function to perform Newton's method on
        df   - the derivative of f
        init - initialization for Newton's method
        eps  - short for epsilon, which is our tolerance level
        N    - how many iterations before we decide the sequence diverges
    """

    class ConvergenceError(Exception):
        pass

    # Newton's method is a special case of the fixed-point method for the following function
    def g(x):
        return x - f(x)/df(x)

    p = [init]
    n = 0
    while n < 100:
        # Calculate p[n+1]
        p.append(g(p[n]))

        # Check for convergence
        if abs(p[n] - p[n+1]) < eps:
            return p
        n = n + 1

    # p[100] doesn't exist
    if abs(p[n-1] - p[n-2]) >= eps:
        raise ConvergenceError("Sequence does not converge")

# f(x) = x - 4x + 3
def f(x):
    return x**2 - 4*x + 3

# f'(x) = 2x - 4
def df(x):
    return 2*x - 4

# Start Newton's method
p = newton(f,df,1.99,10**-5)
for i in range(0,len(p)):
    if p[i] > 0:
        print(f"{i}  {p[i]}")
    else:
        print(f"{i} {p[i]}")
```

The outputs are the following:

n	p_n
0	1.99
1	−48.004 999 999 999 946
2	−23.012 499 000 099 96
3	−10.526 239 505 846 604
4	−4.303 035 962 394 358
5	−1.230 844 833 048 118 3
6	0.229 819 300 167 027 23
7	0.832 452 610 501 521 8
8	0.987 978 163 464 605 4
9	0.999 928 596 128 825 3
10	0.999 999 997 450 925 6
11	0.999 999 999 999 999 9

n	q_n
0	2.01
1	52.005 000 000 001 075
2	27.012 499 000 100 526
3	14.526 239 505 846 885
4	8.303 035 962 394 498
5	5.230 844 833 048 186
6	3.770 180 699 833 003
7	3.167 547 389 498 488 5
8	3.012 021 836 535 395 5
9	3.000 071 403 871 174
10	3.000 000 002 549 074
11	3.0

We see that $\{p_n\}_{n\geq 1}$ converges to 1, whereas $\{q_n\}_{n\geq 1}$ converges to 3.

4 Let $f(x) = 3x - e^x$. Numerically implement the secant method to solve $f(x) = 0$ on $[1, 2]$ with $p_0 = 1$ and $p_1 = 2$. (Does this converge? If not, you may choose another pair of initial p_0 and p_1 you like that leads to convergence.) You may stop the iteration when $|p_N - p_{N-1}| < 10^{-5}$.

Solution My code was written in Python.

```python
def secant(f, init, eps, N=100):
    """
    Implementation of the secant method
    Parameters:
        f    - function to perform the secant method on
        df   - the derivative of f
        init - initialization for the secant method (list with 2 elements)
        eps  - short for epsilon, which is our tolerance level
        N    - how many iterations before we decide the sequence diverges
    """

    class ConvergenceError(Exception):
        pass

    p = init
    n = 1
    while n < 100:
        # Calculate p[n+1] from p[n] and p[n-1]
        p.append(
            p[n] - f(p[n])*(p[n]-p[n-1]) / (f(p[n])-f(p[n-1]))
        )

        # Check for convergence
        if abs(p[n] - p[n+1]) < eps:
            return p
        n = n + 1

    # p[100] doesn't exist
    if abs(p[n-1] - p[n-2]) >= eps:
        raise ConvergenceError("Sequence does not converge")

# f(x) = x - 4x + 3
def f(x):
    return x**2 - 4*x + 3

# Start iteration
p = secant(f,[1,2],10**-5)
for i in range(0,len(p)):
    if p[i] > 0:
        print(f"{i}  {p[i]}")
    else:
        print(f"{i} {p[i]}")
```

The output is the following:

n	p_n
0	1
1	2
2	1.168 615 339 917 483 5
3	1.311 516 554 717 573 3
4	1.797 043 009 631 244 4
5	1.436 777 892 533 490 4
6	1.486 766 286 872 611 7
7	1.515 325 760 523 087 9
8	1.512 011 934 333 299
9	1.512 133 976 002 281 6
10	1.512 134 551 762 062 1