Q1

a): A: $0.0017182 \rightarrow 1.72 \times 10^{-3}$

R: $6.3208 \times 10^{-4} \rightarrow 6.32 \times 10^{-4}$

b): A: $-2.8183 \times 10^{-4} \rightarrow -2.82 \times 10^{-4}$

R: $-1.0368 \times 10^{-4} \rightarrow -1.04 \times 10^{-4}$

c): A: $1.5410 \times 10^{-9} \rightarrow 1.54 \times 10^{-9}$

R: $5.6688 \times 10^{-10} \rightarrow 5.67 \times 10^{-10}$

Q2

a): 4.26×10^{0}

b): 6.45×10^{1}

c): 5.61×10^{1}

d): -3.77×10^{6}

e): 7.51×10^{12}

f): 8.80×10^{2}

g): 2.60×10^{-4}

h): 0.12×10^{-20}

i): 0

j): - Inf

Q3

a): condition number

$= \dfrac{f'(\tilde{x})\,x}{f(x)}$ ($\tilde{x}$ between x, $x+dx$)

$\approx \dfrac{f'(x)\,x}{f(x)}$ (as $dx \to 0$)

$= \dfrac{(1/x)\,x}{\log_e x}$

$= \dfrac{1}{\log_e x}$

① Since $\lim\limits_{x \to 1} \left| \dfrac{1}{\log_e x} \right| = \infty$

Then ill- conditioned

② Since $\lim\limits_{x \to 10} \left| \dfrac{1}{\log_e x} \right| = \left| \dfrac{1}{\log_e 10} \right| < 1$

Then well- conditioned

b):

Code:

```
x1 = 1;
x2 = 10;
dx = 10^(-10);
y1 = log(x1);
y2 = log(x2);
y1_hat = log(x1 + dx);
y2_hat = log(x2 + dx);
condition1 = abs((y1_hat - y1)/(y1))/abs(dx/x1);
condition2 = abs((y2_hat - y2)/(y2))/abs(dx/x2);
fprintf('condition1: %f\n', condition1);
fprintf('condition2: %f\n', condition2);
```

Output:

```
condition1: Inf
condition2: 0.434295
```

explaination.

set $dx = 10^{-10}$

compute condition number for $x=1$, $x=10$ seperately.

computed results agree with prediction

Q4 a): Let $x = 2^{-54}$

Let $a = \frac{1}{1-x} - \frac{1}{1+x}$

$$= \frac{2x}{1-x^2}$$

$$= \frac{2(2^{-54})}{1-2^{-108}}$$

$$= \frac{2^{-53}}{1-2^{-108}}$$

Relative error

$$= \left[fl\left(\frac{1}{1-x} - \frac{1}{1+x}\right) - a \right] / a$$

$$= \left[fl\left(\frac{1}{fl(1-x)} - \frac{1}{fl(1+x)}\right) - a \right] / a$$

$$= \left[fl\left(\frac{1}{1} - \frac{1}{1}\right) - a \right] / a$$

$$\left(\begin{array}{l} fl(1-x) = fl(1-2^{-54}) = 0 \\ fl(1+x) = fl(1+2^{-54}) = 0 \end{array} \right)$$

$$= (0 - a)/a$$

$$= -1$$

which is a large relative error

b): $\frac{1}{1-x} - \frac{1}{1+x} = \frac{2x}{(1-x)(1+x)}$

Assume $x \neq \pm 1$, No overflow, underflow

$$fl\left(\frac{2x}{(1-x)(1+x)}\right)$$

$$= \frac{2x(1+\delta_1)}{(1-x)(1+\delta_2)(1+x)(1+\delta_3)}$$

where $|\delta_1|, |\delta_2|, |\delta_3| \leq \frac{1}{2}\varepsilon_{mach} = \frac{1}{2}2^{-52}$

$$= \frac{2x}{(1-x)} \cdot \frac{(1+\delta_1)}{(1+\delta_2)(1+\delta_3)}$$

$$= \frac{2x}{(1-x)} \cdot \left[1 + \frac{(\delta_1 - \delta_2 - \delta_3 - \delta_2\delta_3)}{(1+\delta_2+\delta_3+\delta_2\delta_3)} \right]$$

$|\delta|$

$$= \left| \frac{(\delta_1 - \delta_2 - \delta_3 - \delta_2\delta_3)}{(1+\delta_2+\delta_3+\delta_2\delta_3)} \right|$$

Since $0 \leq |\delta_1 - \delta_2 - \delta_3 - \delta_2\delta_3| \leq |\delta_1| + |\delta_2| + |\delta_3| + |\delta_2\delta_3|$
and $0 \leq \big| 1 - |\delta_1 + \delta_2 + \delta_2\delta_3| \big| \leq |1 + \delta_2 + \delta_3 + \delta_2\delta_3|$

$$\leq \frac{|\delta_1| + |\delta_2| + |\delta_3| + |\delta_2\delta_3|}{\big| 1 - |\delta_2 + \delta_3 + \delta_2\delta_3| \big|}$$

$$\leq \frac{|\delta_1| + |\delta_2| + |\delta_3| + |\delta_2\delta_3|}{\big| 1 - |\delta_2| - |\delta_3| - |\delta_2\delta_3| \big|}$$

$$\leq \frac{(3/2)\cdot 2^{-52} + (1/4)\cdot 2^{-104}}{1 - 2^{-52} - (1/4)\cdot 2^{-104}}$$

small relative error

Q5 a):

Code1:

```
for x = -25:25
    RE = (exp1(x) - exp(x))/exp(x);
    fprintf('x: %i; relative error: %.10f\n', x, RE);
end
```

Code2:

```
function y = exp1(x)
old_sum = -1;
new_sum = 0;
k = 0;
while old_sum ~= new_sum
    old_sum = new_sum;
    new_sum = new_sum + (x^k)/factorial(k);
    k = k+1;
end
y = new_sum;
```

Output:

```
x: -25; relative error: 58226.1870235171
x: -24; relative error: 9966.3506975240
x: -23; relative error: 66.2289960203
x: -22; relative error: -115.0736551643
x: -21; relative error: 35.3865152208
x: -20; relative error: 1.0249036530
x: -19; relative error: -0.5441982591
x: -18; relative error: 0.0494796394
x: -17; relative error: 0.0010196255
x: -16; relative error: 0.0002852595
x: -15; relative error: 0.0000103538
x: -14; relative error: -0.0000086112
x: -13; relative error: -0.0000012997
x: -12; relative error: 0.0000000612
x: -11; relative error: 0.0000000765
x: -10; relative error: -0.0000000072
x: -9; relative error: -0.0000000005
x: -8; relative error: -0.0000000001
x: -7; relative error: 0.0000000000
x: -6; relative error: -0.0000000000
x: -5; relative error: 0.0000000000
x: -4; relative error: 0.0000000000
x: -3; relative error: 0.0000000000
x: -2; relative error: 0.0000000000
x: -1; relative error: 0.0000000000
x: 0; relative error: 0.0000000000
x: 1; relative error: 0.0000000000
x: 2; relative error: -0.0000000000
x: 3; relative error: -0.0000000000
x: 4; relative error: 0.0000000000
x: 5; relative error: -0.0000000000
x: 6; relative error: 0.0000000000
x: 7; relative error: -0.0000000000
x: 8; relative error: -0.0000000000
x: 9; relative error: 0.0000000000
x: 10; relative error: -0.0000000000
x: 11; relative error: 0.0000000000
x: 12; relative error: -0.0000000000
x: 13; relative error: -0.0000000000
x: 14; relative error: 0.0000000000
x: 15; relative error: 0.0000000000
x: 16; relative error: 0.0000000000
x: 17; relative error: 0.0000000000
x: 18; relative error: 0.0000000000
x: 19; relative error: -0.0000000000
x: 20; relative error: -0.0000000000
x: 21; relative error: -0.0000000000
x: 22; relative error: 0.0000000000
x: 23; relative error: 0.0000000000
x: 24; relative error: 0.0000000000
x: 25; relative error: 0.0000000000
```

b):

for x close to 0, exp1 produces accurate approximation
for x close to -25, exp1 produces poor approximation

when $x \geq 0$, we have positive series,
Each summation produces a small rounding error.
As long as no overflow, accurate approximation.

when $x < 0$, we have alternate series
Since final result $e^x < 1$, while intermediate term $x^k/k!$ may be very large.
Then we get "catastrophic (subtractive) cancellation"

when x close to -25
Then final result is very small, while intermediate term $(-25)^k/k!$ can go up to 5.7×10^9
Then cancellation is more significant
Then large error

c):

Code1:

```
for x = -25:25
    RE = (exp2(x) - exp(x))/exp(x);
    fprintf('x: %i; relative error: %.10f\n', x, RE);
end
```

Code2:

```
function y = exp2(x)
x = x/10; old_sum = -1; new_sum = 0; k = 0;
while old_sum ~= new_sum
    old_sum = new_sum;
    new_sum = new_sum + (x^k)/factorial(k);
    k = k+1;
end
y = new_sum^10;
```

Output:

```
x: -25; relative error: 0.0000000000
x: -24; relative error: 0.0000000000
x: -23; relative error: -0.0000000000
x: -22; relative error: -0.0000000000
x: -21; relative error: -0.0000000000
x: -20; relative error: 0.0000000000
x: -19; relative error: 0.0000000000
x: -18; relative error: 0.0000000000
x: -17; relative error: 0.0000000000
x: -16; relative error: 0.0000000000
x: -15; relative error: -0.0000000000
x: -14; relative error: 0.0000000000
x: -13; relative error: -0.0000000000
x: -12; relative error: -0.0000000000
x: -11; relative error: -0.0000000000
x: -10; relative error: 0.0000000000
x: -9; relative error: 0.0000000000
x: -8; relative error: -0.0000000000
x: -7; relative error: 0.0000000000
x: -6; relative error: -0.0000000000
x: -5; relative error: -0.0000000000
x: -4; relative error: -0.0000000000
x: -3; relative error: -0.0000000000
x: -2; relative error: -0.0000000000
x: -1; relative error: 0.0000000000
x: 0; relative error: 0.0000000000
x: 1; relative error: -0.0000000000
x: 2; relative error: -0.0000000000
x: 3; relative error: -0.0000000000
x: 4; relative error: -0.0000000000
x: 5; relative error: -0.0000000000
x: 6; relative error: 0.0000000000
x: 7; relative error: 0.0000000000
x: 8; relative error: 0.0000000000
x: 9; relative error: 0.0000000000
x: 10; relative error: 0.0000000000
x: 11; relative error: 0.0000000000
x: 12; relative error: -0.0000000000
x: 13; relative error: -0.0000000000
x: 14; relative error: -0.0000000000
x: 15; relative error: 0.0000000000
x: 16; relative error: 0.0000000000
x: 17; relative error: -0.0000000000
x: 18; relative error: -0.0000000000
x: 19; relative error: 0.0000000000
x: 20; relative error: -0.0000000000
x: 21; relative error: 0.0000000000
x: 22; relative error: -0.0000000000
x: 23; relative error: -0.0000000000
x: 24; relative error: -0.0000000000
x: 25; relative error: -0.0000000000
```