1.) We know $x > 0$ as relative error < 1

If $x \leq .2345$ then

$$\text{relative error} = \frac{.2345 - x}{x} \leq 0.002$$

$$\Rightarrow .2345 \leq 1.002x$$

$$\Rightarrow x \geq \frac{.2345}{1.002}$$

If $x > .2345$

$$\text{relative error} = \frac{x - .2345}{x} \leq .002$$

$$\Rightarrow .998x \leq .2345$$

$$\Rightarrow x \leq \frac{.2345}{.998}$$

$$\boxed{\frac{.2345}{1.002} \leq X \leq \frac{.2345}{.998}}$$

③ • The Maclaurin Series for $f(x) = e^{2x}$ is the Taylor series for f at $c = 0$.

As
$$f'(x) = 2e^{2x}$$
$$f''(x) = 4e^{2x}$$
$$\vdots$$
$$f^{(k)}(x) = 2^k e^{2x} \Rightarrow f^{(k)}(0) = 2^k$$

• By Taylor's Theorem, if $x \in \mathbb{R}$ and $n \in \mathbb{N}_0$

$$f(x) = \underbrace{\sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k}_{\text{Maclaurin Series truncated at } x^n} + \underbrace{\frac{f^{(n+1)}(\beta)}{(n+1)!} x^{n+1}}_{\text{Error}}$$

where $0 < \beta < x$.

• When $x = 1$, we have

$$e^2 = f(1) = \sum_{k=0}^{n} \left(\frac{2^k}{k!}\right)(1)^k + \frac{2^{n+1} e^{2\beta}}{(n+1)!}(1)^{n+1}$$

• As $0 < \beta < 1$,

$$\text{error} = \frac{2^{n+1} e^{2\beta}}{(n+1)!} < \frac{2^{n+1} e^2}{(n+1)!} < 10^{-15}$$

this is true when $n \geq 22$

So atleast $\boxed{23 \text{ terms}}$ are needed to compute e^2 with an accuracy of 15 decimal places.

2.)

$$\text{Relative Error} = \left| \frac{13.2399 - 13.24}{13.2399} \right|$$

$$= \left| \frac{-10^{-4}}{13.2399} \right|$$

$$= (13.2399)^{-1}(10)^{-4} \approx 7.55 \times 10^{-6}$$

```python
from math import pi, sin, cos


def main():
    x_0 = pi / 3
    f = sin

    for h in [1e-5, 1e-12]:
        y_approx = (f(x_0 + h) - f(x_0)) / h
        y_exact = cos(x_0)
        rel_error = abs((y_exact - y_approx) / y_exact)
        print(f"h:{h}\ny_approx = {y_approx}\ny_exact = {y_exact}\nrel_error={rel_error}\n")

    print("The fact that the relative error between the actual derivative of the function and the approximated derivative of the function after we increased the st


if __name__ == '__main__':
    main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\MATH 3620> & "c:/MATH 3620/.venv/Scripts/Activate.ps1"
(.venv) PS C:\MATH 3620> & "c:/MATH 3620/.venv/Scripts/python.exe" "c:/MATH 3620/HW1_P4.py"
h:1e-05
y_approx = 0.499995669867026
y_exact = 0.5000000000000001
rel_error=8.660265948257083e-06

h:1e-12
y_approx = 0.5000444502911705
y_exact = 0.5000000000000001
rel_error=8.890058234078956e-05

The fact that the relative error between the actual derivative of the function and the approximated derivative of the function after we increased the step size goes ag
ainst my intuition that as the step sizes get closer, so will the approximation
```

```python

def main():
    getcontext().prec = 36
    a = Decimal(0)
    b = Decimal(1)
    c = Decimal(1 / sqrt(2))
    d = Decimal(0.25)
    e = Decimal(1)
    pi = Decimal(3.14159265358979323846264338327950288)

    print("k\t| f\t\t\t\t\t\t| abs(f-pi)\t\t\t\t\t| g\t\t\t\t\t| abs(g-pi)")

    for k in range(5):
        getcontext().prec = 20
        a = Decimal (b)
        b = Decimal ( (b+c) / 2 )
        c = Decimal.sqrt(c*a)
        d = Decimal ( d - e*(b-a)**2 )
        e = Decimal( 2*e )
        f = Decimal( b**2 / d )
        g = Decimal( (b+c)**2 / (4*d) )
        print(f"{k}\t| {f}\t\t| {abs(f-pi)}\t| {g}\t|{abs(g-pi)}")

    if abs(f-pi) < abs(g-pi):
     print("Convergence: f converges faster on pi than g")
    else:
     print("Convergence: g converges faster on pi than f")

     print("Accuracy: Both f and g achieve 15 significant decimal digits (the maximum precision double-precision floating-point numbers can achieve)")

if __name__ == '__main__':
    main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
(.venv) PS C:\MATH 3620> & "c:/MATH 3620/.venv/Scripts/python.exe" "c:/MATH 3620/HW1_P5.py"
k       | f                                     | abs(f-pi)                                     | g                                     | abs(g-pi)
0       | 3.1876726427121085211         | 0.046079989122315405102       | 3.1405792505221681200 |0.0010134030676249959980
1       | 3.1416802932976531667         | 0.000087639707860050702037    | 3.1415926462135421550 |7.3762509609979634685E-9
2       | 3.1415926538954463688         | 3.0565325280203653146E-10     | 3.1415926535897931111 |4.8979634685441851616E-18
3       | 3.1415926535897931112         | 4.7979634685441851616E-18     | 3.1415926535897931112 |4.7979634685441851616E-18
4       | 3.1415926535897931112         | 4.7979634685441851616E-18     | 3.1415926535897931112 |4.7979634685441851616E-18
Convergence: g converges faster on pi than f
Accuracy: Both f and g achieve 15 significant decimal digits (the maximum precision double-precision floating-point numbers can achieve)
(.venv) PS C:\MATH 3620> 
```