Names: ______________________ ______________________

______________________ ______________________

Teamwork (5)	Discussion (5)	Completeness (5)	Correctness (5)	Total (20)

Measuring the Earth

By Moon or by Sun

I shall be found,

Yet I am undone

If there's no light around.

What am I?

____

Pre-Lab Quiz

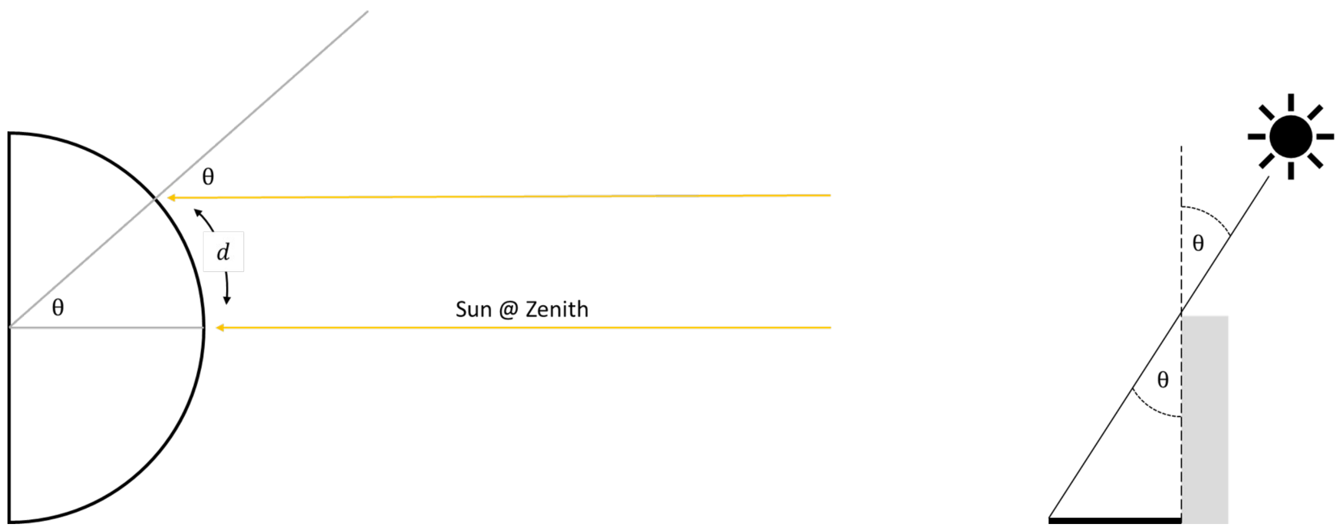
Record you team's answers as well as your reasoning and explanations.

1.
2.
3.
4.

Part 1: The Earth's Radius

In this part we'll estimate the radius of the Earth utlizing the method that Eratosthenes used more than 2000 years ago.

Consider the figure on the left below. There will be some location on Earth such that the Sun is directly overhead. Suppose we are some distance d away from that location. Now the Sun won't be directly overhead at our location, but inclined at an angle θ to our zenith. To measure this inclination we simply need to measure the angle cast by a shadow, as illustrated in the figure on the right.



The distance and angle are directly related to one another, namely that both represent an equivalent fraction of a circle. The distance d represents the same fractional value of the circumference that that angle θ makes with respect to the number of angular units in a circle. In radians (2π radians = 360°) we have the relationship

$$\frac{\theta_{\mathrm{rad}}}{2\pi} = \frac{\text{distance}}{\text{circumference}} = \frac{d}{2\pi r_e} \tag{1}$$

where r_e is the radius of the Earth. So if we measure the angle of the shadow cast by the Sun and know the distance to where on the Earth the Sun is directly overhead we can estimate the size of the Earth.

For night labs, this same method works when using the Moon.

1. Estimate the radius, diameter, and circumference of the Earth in kilometers. Explain how you went about solving this problem. For getting the distance d,

- Go to https://rl.se/sub-solar-point and record the location where the Sun/Moon is at the zenith (e.g., "41.66N, 91.53W").
- On *Google Maps* select "Direction", then enter Iowa City in one cell and the location of the celestial object in the other.
    - Right-click on one of the locations and select "Measure distance"; then left-click on the other location to measure the distance between the two points.

Part 2: The Earth's Mass

You've likely heard the expression `force = mass` $\times$ `acceleration`, or $F = m \cdot a$. In this problem we'll use Newton's law of gravity to estimate the mass of the Earth. Newton's law says that the gravitational force F_g exerted on an object of mass m by a much more massive object with mass $M \gg m$ is given by

$$F_g = m \cdot a = \frac{GMm}{r^2} \tag{2}$$

where $G = 6.67 \times 10^{-11}\ \mathrm{m^3\ kg^{-1}\ s^{-2}}$ is the gravitational constant and r is the distance between the less massive object and the center of the more massive one. For Earth, let $g = a$ denote the acceleration experience by an object on the surface of Earth. Canceling the m term from both sides and solving for the mass of the Earth we find

$$M_{\text{earth}} = \frac{g \cdot r_{\text{earth}}^2}{G} \tag{3}$$

If we know the radius of Earth (which we found in Part 1) and the gravitational acceleration g experienced at its surface we can find its mass. Thus we need some way to measure the gravitational acceleration. Enter the pendulum.

Perhaps you've seen a Foucault Pendulum hanging from the ceiling of a museum, which slowly traces out a circle over the course of a day due to the Earth's rotation. The oscillation period P of one back-and-forth motion of a pendulum depends on the length L of the pendulum and ... the gravitational force exerted on it ($F = m \cdot g$)! Like before, the mass of the pendulum drops out, and in the case where the oscillation angle is small we can use the small angle approximation to derive

$$P^2 = \frac{4\pi^2}{g} L \tag{4}$$

If we let $y = P^2$ and $x = L$, then one has an equation of the form $y = c \cdot x + b$, where $c = 4\pi^2/g$ is the slope and $b = 0$ the intercept. Thus if we measure the periods for several different pendulum lengths we can solve for the gravitational acceleration g by applying a linear fit to the data and deriving the slope.

We'll be using Python for our calculations, but rather than using a terminal we'll be using an Integrated Development Environment (IDE). Your TA will help you get it set up.

1. Before we dive into performing our experiments, our results are critically dependent on getting accurate measurements of the period. Partnering with another group, discuss some ideas for how we can minimize the error in our period estimation. The TA will ask groups to share their thoughts.

---

2. After measuring the length of your pendulum in meters, run several trials and record the period (in seconds) for each. Besides finding the average period, we'll also want to estimate our uncertainty, which we'll take as the average offset from the mean value, ignoring the sign of the offset (*mean absolute deviation*, or MAD). If we make N observations and our average period is $P_{\rm avg}$, our uncertainty on the period δP is then

$$\delta P = \frac{1}{N} \sum_{i=1}^{N} |P_i - P_{\rm avg}| \tag{5}$$

where P_i is our observed period for trial i. We can calculate our average period and its uncertainty using Python as follows, but make sure to use your experimental values

```
import numpy as np

P  = np.array([1.53, 1.56, 1.61, 1.52, 1.55])
dP = np.mean(abs(P - P.mean()))

print("Average P:", P.mean())
print("Uncertainty:", dP)
```

When you have determined the length, period, and the period uncertainty, write your results up on the board where the TA has indicated.

3. Combining the results from all the groups, we are going to fit a linear model to the data to derive an estimate for g. First, let's import `matplotlib.pyplot` for plotting and the `curve_fit` function from the `scipy` (Scientific Python) library to fit our linear model.

```
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
```

Next store the lengths, periods, and the period uncertainties in three arrays. Make sure to use the class values rather than the examples ones given below.

```
L  = np.array([0.25, 0.50, 0.75, 1.00])
P  = np.array([1.01, 1.39, 1.77, 1.98])
dP = np.array([0.02, 0.03, 0.02, 0.04])
```

Since our model equation is $y = x \cdot c$ (we'll ignore the intercept as we expect it to be zero), we can define our linear model `linmod` as a function that takes as input the x values (our lengths) and the slope c and returns the product like so:

```
def linmod(x, c):
     return x * c
```

We have $y = P^2$, but what are our uncertainties on y? To find its uncertainty we need to use calculus. For those familiar with derivatives, we can take the derivative of y with respect to P to get

$$\frac{dy}{dP} = 2 \cdot P \tag{6}$$

If we let $\delta y = dy$ represent the uncertainty on y and $\delta P = dP$ the uncertainty on the period, then we have the relationship

$$\delta y = 2 \cdot P \cdot \delta P \tag{7}$$

In python we can calculate y and its uncertainties as follows, where `**2` means ^2:

```
y  = P**2                          # ** = raise to power
dy = 2 * P * dP                    # Uncertainty on y
```

To derive the slope and its uncertainty we'll use the `curve_fit` function (to learn more about this function, type `curve_fit?` into the terminal and press enter). This function takes as input several parameters, but we are only concerned with the following:

- `f` — a function that takes as input the `xdata` followed by the parameters of the model (in our case, just the slope)
- `xdata` — the independent variable; in our case the pendulum lengths
- `ydata` — the dependent variable; in our case the periods squared
- `sigma` — the uncertainty on the ydata
- `absolute_sigma` — if true, the true uncertainties of the parameters are returned; otherwise, the uncertainties are relative

`curve_fit` returns two arrays, the first one representing the best fit values and the second one is related to the square of the uncertainties. Thus we can get out best fit slope and its uncertainty as follows:

```
slope, err_sq = curve_fit(
     f = linmod,                       # model function to fit
     xdata = L,                        # Independent data
     ydata = y,                        # Observed values (P^2)
     sigma = dy,                       # Uncertainty on ydata
     absolute_sigma = True             # Return true uncertainty
)

err = np.sqrt(err_sq)                  # Find the slope uncertainty

print("slope =", slope)
print("uncertainty =", err)
```

Before blindly accepting the results, one should visually check the model against the data. We can generate our model predictions for a range of lengths as follows

```
xmod = np.linspace(0, max(L), 100)          # Range of x values
ymod = linmod(xmod, slope)                  # Model prediction
```

Finally, let's plot the model prediction against the observed values and their uncertainties:

```
fig, ax = plt.subplots()
ax.plot(xmod, ymod)                    # Model prediction as line
ax.errorbar(L, y, dy, fmt="o")         # Plot data as circles
ax.set_xlabel("L (meters)")
ax.set_ylabel("P$^2$ (sec$^2$)")
fig.show()
```

If your model looks good, show your plot to your TA and have them mark below.

TA	

4. To find the uncertainty on g from the uncertainty in the slope, one would take the derivative dc/dg to derive

$$\frac{\delta c}{\delta g} = \frac{4\pi^2}{g^2} = \frac{c^2}{4\pi^2} \tag{8}$$

where our slope c and its uncertainty δc have units of s^2/m (seconds-squared per meter) since the slope is the *rise-over-run*, where the rise has units of s^2 (period squared) and the run has units of m (length of pendulum).

Working with another group, calculate the gravitational acceleration g and its uncertainty δg, then use Newton's Law to estimate the mass of the Earth. Show your work on a white board and have your TA mark below when done. For reference, the accepted values are $r_{\text{earth}} = 6.4 \times 10^6$ m, $g = 9.8$ m/s^2 and $M_{\text{earth}} = 6 \times 10^{24}$ kg.

TA	