Names: Jo, Kate, Megan

In [7]:
```
# your import statements
import numpy as np
from matplotlib import pyplot as plt
```

WorkSheet Instructions

Before you begin you should have read and worked through Lab 3.

I recommend that you do this worksheet in a python notebook and share screen. This method does mean one person will do the typing. When complete, email the notebook (preferably as a pdf) to sallen@eoas.ubc.ca

This worksheet is based on Question 2 from the lab

Question A

Compute the condition number for the matrix that arises from the Heat Equation using Dirchlet BC for various values of the number of segments, N, the rod is divided into. Note that N segments means N+1 grid points. Consider values of N between 5 and 50. Remember that python starts counting from 0 (not 1).

(Hint: This will be much easier if you write a small Python function that outputs the matrix for a given value of N.)

The Matrix is:

$$A_1 = \begin{bmatrix} 1 & 0 & & \dots & & & & & 0 \\ 1 & -2 & 1 & 0 & \dots & & & & \\ 0 & 1 & -2 & 1 & 0 & \dots & & & \\ & 0 & 1 & -2 & 1 & 0 & \dots & & \\ & & & & & & & & \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & & & & & \\ & & & \dots & 0 & 1 & -2 & 1 & 0 \\ & & & & \dots & 0 & 1 & -2 & 1 \\ 0 & & & & & \dots & & 0 & 1 \end{bmatrix}$$

In [11]:

```
# function that outputs matrix for a given value of N
N=10 #value between 5 and 50

condition_n = np.zeros(50)


for n in range(5,50):
    A1 = np.zeros([n,n]) #makes everything 0
    A1[0,0] = 1
    for i in range(1,n-1):
        A1[i,i-1] = 1
        A1[i,i] = -2
        A1[i,i+1] = 1
    A1[n-1,n-1] = 1
    condition_n[n] = np.linalg.cond(A1) #Calculating condition number of al
```

In [14]:

```
# cell to calculate the condition number for various N
#See above
print(len(condition_n))
```

```
50
```

Question B

Plot your results on a log-log plot (that is log condition number versus log N)

Also plot N^2 on the same plot.

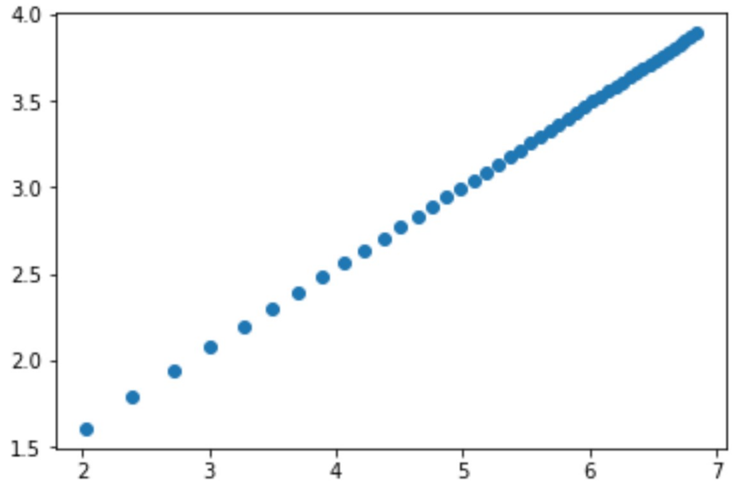
How does the conition number of A_1 depend on N?

In [15]:

```
# code to do the plot
xlog = np.log(condition_n) #This is for all the K's
ylog= np.log(range(0,50))  #this is for all the N's

plt.scatter(xlog,ylog)
plt.show()
```

```
C:\Users\megan\AppData\Local\Temp/ipykernel_8684/2737152065.py:2: RuntimeWar
ning: divide by zero encountered in log
  xlog = np.log(condition_n) #This is for all the K's
C:\Users\megan\AppData\Local\Temp/ipykernel_8684/2737152065.py:3: RuntimeWar
ning: divide by zero encountered in log
  ylog= np.log(range(0,50))  #this is for all the N's
```



Question C

Another way to write the system of equations Another way to write the system of equations is to substitute the boundary conditions into the equations, and thereby reduce size of the problem to one of N-1 equations in N-1 unknowns. The corresponding matrix is simply the N-1 by N-1 submatrix of A_1

$$A_2 = \begin{bmatrix} -2 & 1 & 0 & \cdots & & & 0 \\ 1 & -2 & 1 & 0 & \cdots & & \\ 0 & 1 & -2 & 1 & 0 & \cdots & \\ & & & & & & \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & & & & 0 \\ & & \cdots & 0 & 1 & -2 & 1 \\ 0 & & & \cdots & 0 & 1 & -2 \end{bmatrix}$$

Does this change in the matrix make a significant difference in the condition number?

In [17]:

```
# new function that outputs the matrix A_2 for a given value of N
N = A1.shape[0] # lenght of matrix N

A_2 = A1[1:N-1,1:N-1 ] # subset to make matrix smaller (slice off the first
```

In [4]:

```
# cell to caculate the condition number for various N
condition_n = np.zeros(50)

##Not quite done this one##

for n in range(5,50):
    A1 = np.zeros([n,n]) #makes everything 0
    A1[0,0] = 1
    for i in range(1,n-1):
        A_2[i,i-1] = 1
        A_2[i,i] = -2
        A_2[i,i+1] = 1
    A_2[n-1,n-1] = 1
    condition_n[n] = np.linalg.cond(A_2) #Calculating condition number of a
```

In [ ]:

```
# log log plot comparing
xlog = np.log(condition_n) #This is for all the K's
ylog= np.log(range(0,50))  #this is for all the N's

plt.scatter(xlog,ylog)
plt.show()
```