

```
In [1]: import matplotlib.pyplot as plt
import numpy as np
```

**Question 1.** Assign `x` to the smallest prime number.

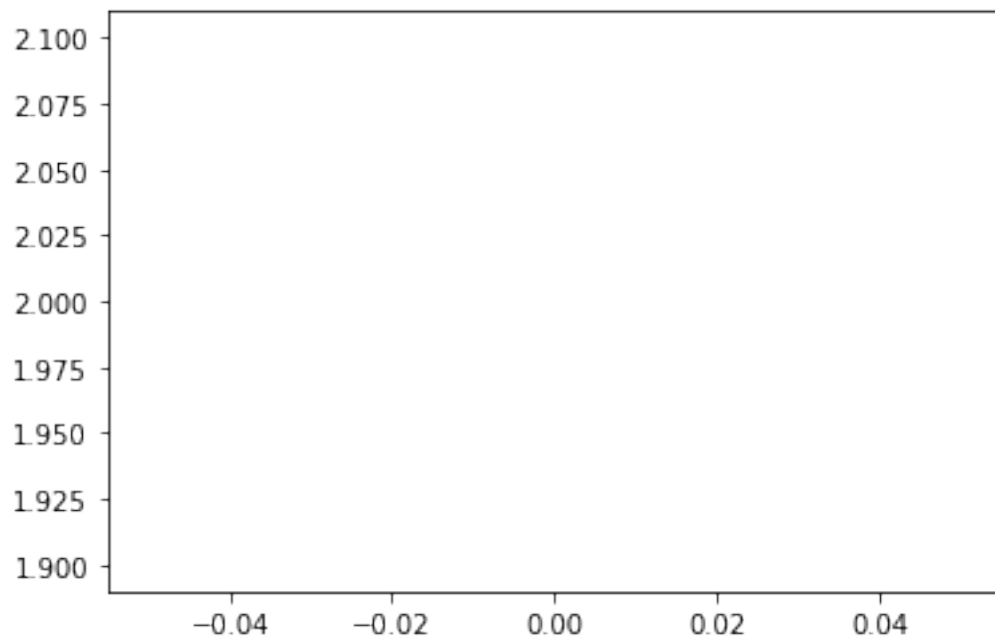
```
In [22]: x = 2 # SOLUTION
```

```
In [ ]: grader.check("q1")
```

**Question 2.** Visualize the answer

```
In [28]: ## solution ##
plt.plot(x); # SOLUTION NO PROMPT
""" # BEGIN PROMPT
plt.plot(...);
""" # END PROMPT
```

```
Out[28]: ' # BEGIN PROMPT\nplt.plot(...);\n'
```



This cell is not part of a question.

```
In [29]: y = 3
```

**Question 3.** Define `square` and assign `nine` to 3 squared.

```
In [30]: def square(x):
          y = x * x # SOLUTION
          return y # SOLUTION

          nine = square(3)
```

```
In [ ]: grader.check("q3")
```

**Question 4.** What does equilateral mean?

*Type your answer here, replacing this text.*

**SOLUTION:** Having equal side lengths.

```
In [14]: # this isn't part of a question
          # it's here to make sure that we get a MD cell above to close the export
          # of question 4
```

**Question 5.** Approximate the area and circumference of a circle with radius 3.

```
In [15]: pi = 3.14
          if True:
              # BEGIN SOLUTION
              radius = 3
              area = radius * pi * pi
              # END SOLUTION
              print('A circle with radius', radius, 'has area', area)

          def circumference(r):
              # BEGIN SOLUTION NO PROMPT
              return 2 * pi * r
              # END SOLUTION
              """ # BEGIN PROMPT
              # Next, define a circumference function.
              pass
              """; # END PROMPT
```

A circle with radius 3 has area 29.5788

```
In [16]: # This question has no tests.
```

**Question 6.** Write something

*This question has a custom prompt below, so that prompt should be in the output. It also has no solution!*

*Write your thing here.*

**SOLUTION:** some thing

**Question 7:** What is the answer?

*Type your answer here, replacing this text.*

**SOLUTION:** 42

**Question 8:** Test intercell seeding by generating 10 random  $N(4, 2)$  numbers.

```
In [35]: np.random.seed(42) # SEED
         z = np.random.normal(4, 2, 10) # SOLUTION
         z
```

```
Out[35]: array([4.99342831, 3.7234714 , 5.29537708, 7.04605971, 3.53169325,
                3.53172609, 7.15842563, 5.53486946, 3.06105123, 5.08512009])
```

```
In [ ]: grader.check("q8")
```

**You're done!**

```
In [ ]:
```

---

To double-check your work, the cell below will rerun all of the autograder tests.

```
In [ ]: grader.check_all()
```

## 0.1 Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**

```
In [ ]: # Save your notebook first, then run this cell to export your submission.  
        grader.export()
```