

Lecture 9 - Exercises

9A - import statements

1. Which of the following programs will **not** cause an error when run?

```
import random
random.randint(6, 10)
```

```
import random
random.randint(10)
```

```
from math import sin
sin(math.pi)
```

```
import math
from math import cos
cos(math.pi)
```

```
import math
from math import cos
math.cos(math.pi)
```

2. What is the largest number that could be printed by the following program?

```
import math
import random

print(math.ceil(random.uniform(random.random()*10, 30)))
```

3. What is the difference between the `random` module's `randint` and `randrange` functions?

9B - Turtles

As a reminder, for both of the following problems, I highly recommend trying to answer them before running the code to verify your answer.

4. Describe in words what the following code draws on the Turtle's canvas:

```
import turtle
t = turtle.Turtle()
for i in range(30):
    t.forward(5)
    t.penup()
    t.forward(5)
    t.pendown()
```

5. After the above code is run, what is its position (in x, y coordinates) on the canvas? You should be able to find the information needed to answer this question in the `turtle` module's documentation.

Problems

1. Write a program that generates and prints random integers between 1 and 10 (inclusive) until one of the random numbers exceeds 8. At the end, print how many numbers were generated. Here's one run of the program (of course, the output will be different each time!).

```
4
5
4
Generated 3 numbers.
```

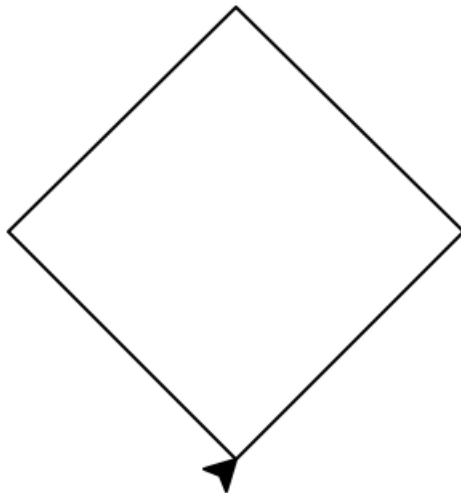
2. Write a program that takes two **command line arguments**. The first is a number (**d**) of sides on a die; the second is a number of such dice (**n**). The program should print the result of rolling **n d**-sided dice. A **d**-sided die has numbers from 1 to **d** and each of them is equally likely to land facing up. For example, here are two outputs for rolling 2 6-sided dice:

```
>>> %Run ndice.py 6 2
1 4
>>> %Run ndice.py 6 2
5 6
```

and here's an example of rolling 4 12-sided dice:

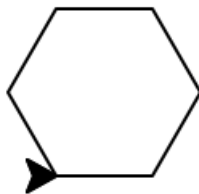
```
>>> %Run ndice.py 12 4
3 9 12 3
```

3. Write a program that creates a turtle object and draws the following picture. This is a square with side length 100, but drawn tilted at a 45-degree angle. The turtle's location in the picture is its starting point, i.e., the middle of the canvas.

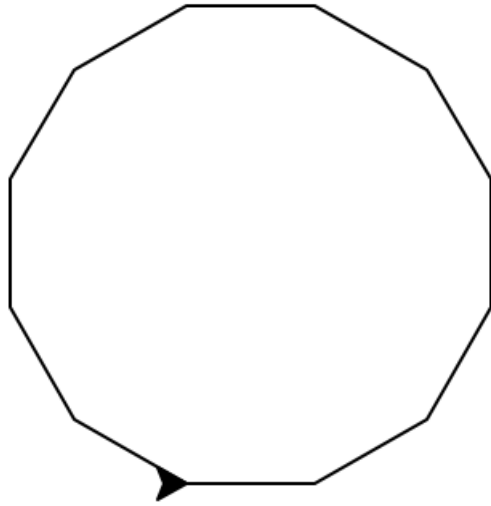


4. Write a program that creates a turtle and uses it to draw an **n**-sided polygon with side length **s**. The program takes **n** and **s** as **command line arguments**. *Hint*: the square I drew in lecture was a 4-sided polygon with side length 100. *Hint 2*: Regardless of the number of sides, the turtle needs to turn a total of 360 degrees while drawing the entire polygon.

Here's the result of drawing a 6-sided polygon (hexagon) with side length 30:



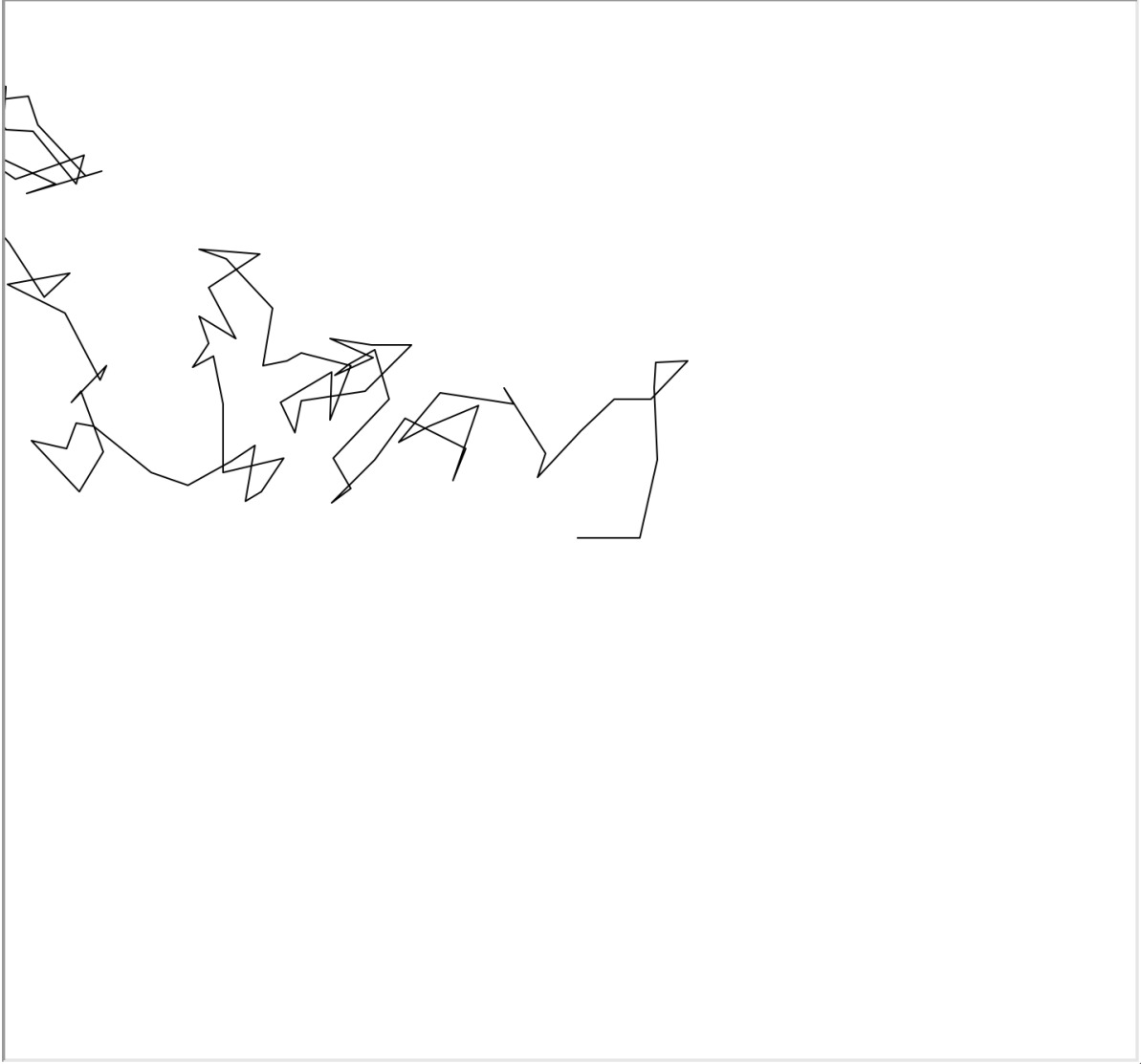
and here's a 12-sided polygon (dodecagon) with side length 40:



5. Write a program to make a turtle do a **random walk**. In other words, repeat the following 100 times:

- Move a random distance forward
- Turn a random number of degrees

I chose a random distance forward between 10 and 50, but you can choose whatever distance you'd like. Here's one output:



If you're feeling fancy, devise a way to modify the program to prevent the turtle from walking off the screen.