1. Consider the following function definition.

\[
\text{def test(a = 10, b = 15, c = 20): return (a/b)*c}
\]

For each of the function calls below, write down (i) the values that the parameters of test take on and (ii) the value that is returned by the function.

(a) test()
(b) test(10.0)
(c) test(30)
(d) test(40, 40)
(e) test(c = 50)
(f) test(c = 50, a = 15)
(g) test(30, c = 50, b = 10)
(h) test(40, 40, 40)
(i) test(len("hello"), 1)
(j) test(round(-8.678))

2. These problems go together and you should submit a single Python file corresponding to Parts (a)-(c) of this problem.

(a) Write a function that simulates the roll of two \(n\)-sided dice and returns the sum of the outcomes of the two dice rolls. The function should take \(n\) as a parameter with a default value of 6.

(b) Write a function that simulates the roll of two \(n\)-sided dice \(m\) times and returns the number of times \(p\) appears as the sum of the outcomes of the two dice rolls. The function should take \(n\), \(m\), and \(p\) as parameters with a default value of 6 for \(n\) and 1000 for \(m\). This function should repeatedly call the function you wrote for Part (a).

(c) Write a program that prints out, for each integer \(i\), \(2 \leq i \leq 12\), the number of times (out of 1000 trials) that \(i\) appears as the sum of the rolls of two 6-sided dice. In other words, your program should roll two 6-sided dice 1000 times and report the number of times 2 shows up. It should then roll two 6-sided dice 1000 times and report the number of times 3 shows up. It should repeat this process for each \(i = 2, 3, 4, \ldots, 12\). Your program should call the function you wrote for Part (b) repeatedly.

(d) Examine the numbers you got in Part (c) for each \(i = 2, 3, 4, \ldots, 12\). Are they all roughly equal, i.e., is each number 2 through 12 equally likely to appear as the sum of the two dice rolls? In 2-3 sentences can you describe and explain the trend you see in this data.
3. Write a function called `twoDRandomWalk` that simulates a 2-dimensional random walk. This function starts off a “robot” at point (0, 0) and then in each step of the random walk the robot moves 1 step in one of 4 directions (north, south, east, west) chosen at random with equal probability. Imagine that there is a $2n \times 2n$ square “barrier” centered around point (0, 0) and the random walk ends when the robot reaches any point on this barrier. More precisely, for any given positive integer $n$, the barrier is defined by the lines $x = n$, $x = -n$, $y = n$, and $y = -n$. The function should return the number of steps the robot took before ending the random walk. The function should have the following header:

```python
def twoDRandomWalk(n = 100, printOn = False):
```
where the first parameter $n$ specifies the barrier, while the second argument `printOn` tells the function whether it should do its work quietly or whether it should print the locations of the robot and it travels. In other words, if `printOn` is `True` then the function prints the robot locations as it moves. This is in addition to returning the length of the random walk. If `printOn` is `False` then the function prints nothing and simply returns the length of the random walk.

4. Extra Credit: 10 points. The 2011 University of Iowa Computing Conference is being held on Feb 25th (starting at 6 pm on Friday) and Feb 26th. For details visit [http://www.acm.uiowa.edu/uicc/](http://www.acm.uiowa.edu/uicc/). The keynote speech is by Prof. Ron Vetter, who will talk about developing mobile phone applications. This is in Room 1505 Seamans Center at 6 pm. Attend this talk and write a 1 paragraph (5-7 sentences) summary of the talk.