**Problem 1: A game of chance, rolling a single die**

In the development of probability theory that underpins much of statistics, the roll of a “fair die” is often cited as a key concept. We want to write a simple program that utilizes a random number generator to mimic the behavior of a fair die.

**To Do:**

1. Create a file called myModule.py in PyCharm.
2. Create a function called rollDie() that takes no arguments and returns an integer between 1 and 6 inclusively, but at random each time the function is called.
3. In a separate file called HW1SP22\_Prob1.py, write a main() function that imports myModule and calls rollDie() 1000 times and tabulates the fraction of rolls that yield 1, 2, 3, etc. Output the probability of each of the possible number to the screen as:

Probability of rolling a 1: 0.1667

Probability of rolling a 2: 0.1668

Etc.

Do these probabilities match the theory? If not, see what happens if we roll the die 10,000 times.

**Problem 2: Rolling dice**

Now that we can roll a single die, we want to see the probabilities of rolling any possible number when we have n dice rolled simultaneously.

**To Do:**

1. Modify your file called myModule.py to include a function called rollDice(N=1), where N is a keyword argument that specifies the number of dice to be rolled simultaneously and returns the total score by summing the score of each die.
2. In a separate file called HW1SP22\_Prob2.py, write a main() function that specifies the number of dice (n) and call the function rollDice(N=n) 1000 times and then outputs the probability for each possible score to the screen as:

Probability of rolling a 3: 0.xxx

Etc.

(note: for 3 dice, min score =3, max score =18, for 4 dice, min score = 4, max score = 24, etc.)

Do the probabilities change if we roll the dice 100 times? 10,000 times?

**Problem 3: A discrete random variable**

**Description:** Often in simulating the behavior of real-world systems, we must produce normally distributed values for a property of a system (e.g., the average mass of a pebble in a pile of gravel). Produce a program that yields an array of data that is normally distributed around a specified mean and with a specified standard deviation. (Note: if I use a random number generator to produce a uniformly distributed value between 0 and 1, we can use this value as the dependent variable on a cumulative distribution function and find the independent variable value (x) corresponding to this probability. The values of x will be normally distributed.) Your program should output an array of size N that is normally distributed. You should do this work in a file called HW1SP22\_Prob3.py

Hint: from the module random, you should investigate the use of the functions: random() and normalvariate()