Homework 6

In this assignment, we will explore the use of random numbers in determining a solution. In this case, we will try to compute pi using an ancient method.

Consider the area of a square with side length of 2 centered on the origin. It has an area of 4 square units.

Now consider the area of a circle with radius 1 centered on the origin. It has an area of πr^2, or just π.

A method to compute pi is to generate two random numbers, x and y, each with a uniform distribution between [0,1). (For our purposes, we can neglect that we can't get 1 since the odds of generating the value 1 are so small.) Compute the value of r. If it is less than or equal to 1, then we're inside the circle. Otherwise, we're outside the circle but inside the square.

If our random number generator is of good quality, and if we ran for long enough, we could get an excellent value of pi by computing the fraction 4*(number of times we were inside)/(total number of trials). Please satisfy yourself that this is true!

The idea is to create a loop that keeps generating random numbers and comparing what you get to np.pi until you are within a certain precision. For this assignment, use the powers of 10 from 10^{-1} to 10^{-7} , inclusive. We'll want the average value of pi computed at each precision for the SUCCESSFUL attempts. (See the next paragraphs.)

To get an average value at each precision, attempt to compute the value 100 times at each precision.

We need to set a limit of how many random points we'll generate before giving up since, for a high enough precision, it could take years to successfully compute pi this way. For this assignment, give up if, after 10,000 points have been generated, you still haven't achieved the required precision. If you give up, do not count this attempt toward the average value. But, it does count as one of the 100 attempts at that precision.

Your assignment is as follows:

Write a Python script that uses the above method:

Compute the average value of pi found for the successful attempts at each precision.

Print your results as shown below. (Your numbers will differ. In fact, your numbers will differ every time your run the script!) Each line shows the precision, the number of successful attempts out of 100 at that precision, and the average value of pi computed at that precision.

```
> python hw6.py
0.1 success 100 times 3.156278862544277
0.01 success 98 times 3.141856263165276
```

0.001 success 96 times 3.14126629328762
0.0001 success 88 times 3.141585673244998
1e-05 success 79 times 3.1415918588627307
1e-06 success 51 times 3.1415927261885632
1e-07 no success