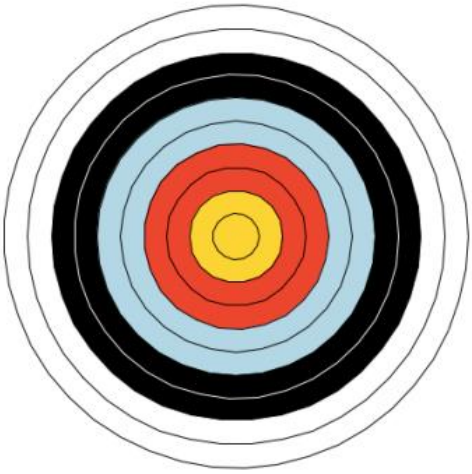
Program: Single-shot Archery

According to http://en.wikipedia.org/wiki/Target_archery, standard FITA targets are marked with 10 evenly spaced concentric rings, which … have score values from 1 through 10 assigned to them. … In FITA archery, targets are colored as follows:



- 1 ring & 2 ring – white
- 3 ring & 4 ring – black
- 5 ring & 6 ring – blue
- 7 ring & 8 ring – red
- 9 ring, 10 ring - gold

a. Write a function called *target* to draw an archery target using turtle graphics; target should have a parameter that determines the size of the drawing, and the center of the target should be at the origin (0, 0).

b. Write a function *shoot* that simulates firing an arrow towards the target. The effect of shoot should be to indicate visually where on the target (or near the target) the "arrow" lands. The landing position should be at a pseudo-random position on or near the target. Look at the documentation for the Python module random and choose a suitable function to help you generate a pseudo-random position. The return value of shoot should be the distance between the origin and the point at which the arrow lands.

c. Write a function *score*, which takes as parameters the distance of an arrow from the origin, and the radius of the target, and calculates the score for that arrow, according to the above instructions. You may need to use some of the functions in the math module. The return value of score should be the calculated score. Put some examples into the docstring for score, to illustrate how the function works, like this:

```
Examples:
>>> score(0, 200)
10
>>> score(250, 200)
0
```

Add enough examples to make clear what score does in the interesting boundary cases. When you execute these examples in the interpreter, you should get the illustrated answers!

d. Write a function *displayScore* that writes a score underneath a target. Give it suitable parameters.

e. Complete your program for playing single-shot archery by writing the function *singleShot* that draws the target, fires an arrow, and displays the score. Naturally, singleShot should use the other functions that you have already defined. Give singleShot any necessary parameters.

f. Test your program by executing singleShot a few times.