Problem 28

Let $k_n[r]$ be the corner numbers of the spiral, where $n \in [0, 3]$ are the four corners at a given radius $r \in [0, \infty)$. We define $k_n[0] = 1$. With this definition, the first few sequences are:

$$
\begin{aligned}
k[0] &= 1 \\
k[1] &= 3, 5, 7, 9 \\
k[2] &= 13, 17, 21, 25 \\
k[3] &= 31, 37, 43, 49
\end{aligned}
$$

It is easy to recognize the pattern that forms and the sequence can be written more generally by:

$$k_n[r] = k_0[r] + 2nr \qquad \forall n \in [0, 3], \quad r \in [1, \infty), \quad k_n[0] = 1 \tag{1}$$

The sequence of $k_0[r]$ values is $k_0[r] = 1, 3, 13, 31, \ldots$. We can write this sequence as the following:

$$k_0[r] = k_3[r-1] + 2r \qquad \forall r \in [1, \infty) \tag{2}$$

Which gives us the general form of the discrete function for the corner numbers:

$$k_n[r] = k_3[r-1] + 2r + 2rn \qquad \forall n \in [0, 3], \quad r \in [1, \infty), \quad \text{with } k_n[0] = 1 \tag{3}$$

With this recurrant sequence definition, the sum of the diagonals of the spiral of radius 500 (1001x1001 square) is given by:

$$S = 1 + \sum_{r=1}^{500} \sum_{n=0}^{3} k_n[r] \tag{4}$$

This is easy to program with recursion. An implementation in python follows.

```
def cornerNum(n,r):
        if not n in range(4):
                print "Invalid value of n"
                return None
        if r < 0:
                print "Invalid value of r"
                return None

        if r == 0:
                return 1

        else:
                return cornerNum(3,r-1) + 2*r + 2*r*n

sum = 1
for r in range(1,501):
        for n in range(0,4):
                sum += cornerNum(n,r)

print sum
```