day44 happy numbers

Today we have two assignments. This is the first one.

Write a program that creates a list of happy numbers. A "happy number" is defined as follows:

> To check if a number is happy, square each digit in the number, add the squares together, and then repeat. After one or more times doing this, your result will either become 1, in which case your initial number is happy, or your results will cycle through a series of numbers forever (specifically: 4, 16, 37, 58, 89, 145, 42, 20, 4, ... ) in which case the number is not happy.

Once again, this is a listing in that crazy list database online: the Online Encyclopedia of Integer Sequences (http://oeis.org/)/ Happy numbers are at this link: http://oeis.org/A007770

For example, if you started with 7, your check to see if 7 is happy would involve the following:

$$7^2 = 49$$
$$4^2 + 9^2 = 97$$
$$9^2 + 7^2 = 130$$
$$1^2 + 3^2 + 0^2 = 10$$
$$1^2 + 0^2 = 1$$
$$1^2 = 1$$

So 7 is a happy number because the process ends with 1. (If we kept going, we'd get 1 over and over again.)

If we try 3, we get a different result:

$$3^2 = 9$$
$$9^2 = 81$$
$$8^2 + 1^2 = 65$$
$$6^2 + 5^2 = 61$$
$$6^2 + 1^2 = 37$$

Because 37 is in the list that repeats forever (4, 16, 37, 58, 89, 145, 42, 20, 4, ...), 3 is not happy.

If you land anywhere in the list of repeating unhappiness, you never escape. The bottom line is that you either end up with 1, or you end up landing on this dead-end list and never get off.

Your program will check the numbers from 1 through 100 and report whether each is happy or unhappy. Your program will use a function to break a number apart and add the squares of its digits.

Write your program so that it can handle any length numbers, for example 1 digit, 2 digit, or 3 digit numbers.

(continued on back)

Requirements:

- Program uses a function called "processDigits" above the main part of your program that breaks apart a number, squares each digit, adds the squares, then returns a number value. We learned how to do functions in day29. Take a look at your day29 assignment for help.
- Program uses a for loop to print and process the numbers from 1 to 100 and finds and reports which are happy and which are unhappy.

Partial sample output:

```
  1: happy
  2: unhappy
  3: unhappy
  4: unhappy
  5: unhappy
  6: unhappy
  7: happy
  8: unhappy
  9: unhappy
 10: happy
  ...
```

For the record, the happy numbers up to 100 are:

1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100

Tips:

- In your processDigits function you need to take the number sent to the function and first convert it to a string: num = str(num) . Then you can square each digit and add all the squares (see next bullet item). Use a return call to send the answer back. This function just squares and adds the digits. It doesn't decide if a number is happy.

- processDigits must use a for loop to go through all of the digits and square them and add them together. In processDigits you will put the number into a string variable, say num = "123". Use a for loop (for x in num) where x would be "1", then "2", then "3" to access each digit, convert it into an integer using int() and then squaring it, adding everything together.

- If you don't remember how to make a function, review online day29. You'll be writing a function that uses the "return" command.

- Your main program should be a for loop that uses the range (1,101). Inside the for loop you will have a while True: loop that sends each number to your function over and over until you either get 1 back, or you land on the list of unhappy numbers (You can just check for 4 as a return value, as that means the number is unhappy.) You break out of the while loop when you know that a number is happy or unhappy.

Extra credit: print the results out in four neat columns that line up:

```
 1: happy        2: unhappy          3: unhappy          4: unhappy
 5: unhappy      6: unhappy          7: happy            8: unhappy
 9: unhappy     10: happy...
```