Happy Numbers

Let the sum of the squares of the digits of a positive integer s_0 be represented by s_1. In a similar way, let the sum of the squares of the digits of s_1 be represented by s_2, and so on. If $s_i=1$ for some $1\leq i$, then the original integer s_0 is said to be **happy**. For example, starting with 7 gives the sequence **7**, **49** (=7^2), **97** (=4^2+9^2), **130** (=9^2+7^2), **10** (=1^2+3^2), **1** (=1^2), so 7 is a happy number, which reaches 1 on 6 iterations.

The first few happy numbers are 1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100, …. The number of iterations i required for these to reach 1 are, respectively, 1, 6, 2, 3, 5, 4, 4, 3, 4, 5, 5, 3, ... A number that is not happy is called **unhappy**. Once it is known whether a number is happy (unhappy), then any number in the sequence s_1, s_2, s_3, ... will also be happy (unhappy). Unhappy numbers have eventually periodic sequences of s_i which do not reach 1 (e.g., 4, 16, 37, 58, 89, 145, 42, 20, 4, ...). You need to write a program to find all the happy numbers in a given closed interval, which reach 1 within 10 iterations.

Input Specification

- It is a single line input of two positive integers separated by a space.

Output Specification

- Print all happy numbers in the interval and the number of iterations required by it to reach 1, separated by a space and each in a new line.

Sample Input 7 11 **Sample Output** 7 6 10 2	**Sample Input** 44 68 **Sample Output** 44 5 49 5 68 3