Problem D. Happy Prime Number

A *Happy Number* ☺ can be defined as follows. From a positive integer *n*, calculate the sum of square of each digit of *n*. Then from that sum, repeat the same process over and over again. This cycle terminates if and only if there is 1 in the sequence. Hence, we call the number *n* a *happy number* if it generates a finite sequence. Otherwise, the endless cycle occurs (1 never appears in the sequence). We may call the number generating an endless cycle an unhappy number ☹. Observe the following examples:

700 **is** a happy number

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

$1^2 + 3^2 + 0^2 = 10$ $1^2 + 0^2 = 1$

2 **is not** a happy number

$2^2 = 4$ $4^2 = 16$ $1^2 + 6^2 = 37$

$3^2 + 7^2 = 58$ $5^2 + 8^2 = 89$ $8^2 + 9^2 = 14$

$1^2 + 4^2 + 5^2 = 42$ $4^2 + 2^2 = 20$ $2^2 + 0^2 = 4$

$4^2 = 16$ ... and never terminates

A *Prime Number* is an integer greater than 1 that can be divided only by 1 and itself. Here are some prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, …

A *Happy Prime Number* is a prime number which also satisfies the happy number condition such as 7, 13, 19, …

Your task is to write a program to show all happy prime numbers less than or equal to a given number *n*. ($10 \leq n \leq 1000000$)

Input and Output

A positive number *n* is the only input of the program and the program prints all happy numbers in ascending order, one number in a line.

Sample Input	**Sample Output**
20	7 13 19