

6517 Federation Favorites

En route to Rigel 7, Chief Engineer Geordi Laforge and Data were discussing favorite numbers. Geordi exclaimed he preferred Narcissistic Numbers: those numbers whose value is the same as the sum of the digits of that number, where each digit is raised to the power of the number of digits in the number.

Data agreed that Narcissistic Numbers were interesting, but not as good as his favorite: Perfect Numbers. Geordi had never heard of a Perfect Number, so Data elaborated, “A positive integer is said to be Perfect if it is equal to the sum of its positive divisors less than itself. For example, 6 is Perfect because $6 = 1 + 2 + 3$.”

Geordi began thinking about an algorithm to determine if a number was Perfect, but did not have the raw computing ability of Data. He needs a program to determine if a given number is Perfect.

Help Geordi write that program.

Input

Input consists of a single entry per line. Each line contains a single positive integer n , where $2 < n < 100,000$ for each case. A line containing ‘-1’ denotes the end of input and should not be processed.

Output

For each case, determine whether or not the number is Perfect. If the number is Perfect, display the sum of its positive divisors less than itself. The ordering of the terms of the sum must be in ascending order. If a number is not Perfect, print ‘< NUM > is NOT perfect.’ where < NUM > is the number in question. There must be a single space between any words, symbols, or numbers in all output, with the exception of the period at the end of the sentence when a number is not perfect.

Sample Input

```
6
12
28
-1
```

Sample Output

```
6 = 1 + 2 + 3
12 is NOT perfect.
28 = 1 + 2 + 4 + 7 + 14
```