

7348 Talking About Numbers

The programmers of a TTS (text-to-speech) program have discovered that their product does a poor job of reading numbers written in conventional numeric form. Currently, it just announces each digit in the number, one after the other, which gets confusing to the listener after a few digits have gone by.

They would prefer to have a more natural reading of numbers, and one of them has suggested that, if they convert numeric strings into the appropriate text equivalent, e.g., convert “1023” into “one thousand and twenty three”, then their speech engine will be able to handle the reading with no further modification.

Write a program to carry out the transformation of non-negative integers into conventional English wording:

- English has unique names for the numbers 0-19: “zero”, “one”, “two”, “three”, “four”, “five”, “six”, “seven”, “eight”, “nine”, “ten”, “eleven”, “twelve”, “thirteen”, “fourteen”, “fifteen”, “sixteen”, “seventeen”, “eighteen”, “nineteen”.
- The subsequent multiples of 10 are named “twenty”, “thirty”, “forty”, “fifty”, “sixty”, “seventy”, “eighty”, “ninety”.
- The combination of one of those multiples of ten with a digit 1-9 is always hyphenated: e.g., 31 \Rightarrow “thirty-one”, 77 \Rightarrow “seventy-seven”.
- Multiples of 100 are counted 1-9 and set off from any following non-zero digits by ‘and’: e.g., 200 \Rightarrow “two hundred”, 412 \Rightarrow “four hundred and twelve”, 777 \Rightarrow “seven hundred and seventy-seven”.
- Thousands and millions are counted off using the above rules to form numbers 1-999, and are set off from any non-zero remainder by a comma: e.g., 1,253,101 \Rightarrow “one million, two hundred and fifty-three thousand, one hundred and one”.
- If a number with a non-empty thousands or millions component is followed by a remainder of 1-99, then instead of a comma the parts are separated by “and”: e.g., 1,000,011 \Rightarrow “one million and eleven”, 20,222,043 \Rightarrow “twenty million, two hundred and twenty-two thousand and forty-three”.

Input

Input will consist of one or more datasets. Each dataset will consist of a single line containing a non-negative integer in the range 0...999,999,999. Although we have used commas within digit strings for clarity in this problem description, there will be no commas in the input. There will be no leading zeros on positive input numbers.

A line with a negative value signals the end of input.

Output

For each dataset, print a single line containing the spelled-out equivalent of the number, according to the rules above.

Formatting requirements:

- The output must be left-justified.
- All alphabetic characters must be in lower-case.

- Exactly one blank must separate adjacent words, except when a hyphen or comma is called for.
- When a comma is used, it must be followed by exactly one blank.
- When a hyphen is used, no blank space appears to either side of the hyphen.

Sample Input

```
1
222143
1000001
-1
```

Sample Output

```
one
two hundred and twenty-two thousand, one hundred and forty-three
one million and one
```