There are n people standing in a line, playing a famous game called “counting”. When the game begins, the leftmost person says “1” loudly, then the second person (people are numbered 1 to n from left to right) says “2” loudly. This is followed by the 3rd person saying “3” and the 4th person say “4”, and so on. When the n-th person (i.e. the rightmost person) said “n” loudly, the next turn goes to his immediate left person (i.e. the $(n-1)$-th person), who should say “$n+1$” loudly, then the $(n-2)$-th person should say “$n+2$” loudly. After the leftmost person spoke again, the counting goes right again.

There is a catch, though (otherwise, the game would be very boring!): if a person should say a number who is a multiple of 7, or its decimal representation contains the digit 7, he should clap instead! The following tables shows us the counting process for $n = 4$ (‘X’ represents a clap). When the 3rd person claps for the 4th time, he's actually counting 35.

Person	1	2	3	4	3	2	1	2	3
Action	1	2	3	4	5	6	X	8	9
Person	4	3	2	1	2	3	4	3	2
Action	10	11	12	13	X	15	16	X	18
Person	1	2	3	4	3	2	1	2	3
Action	19	20	X	22	23	24	25	26	X
Person	4	3	2	1	2	3	4	3	2
Action	X	29	30	31	32	33	34	X	36

Given n, m and k, your task is to find out, when the m-th person claps for the k-th time, what is the actual number being counted.

Input

There will be at most 10 test cases in the input. Each test case contains three integers n, m and k ($2 \le n \le 100$, $1 \le m \le n$, $1 \le k \le 100$) in a single line. The last test case is followed by a line with $n = m = k = 0$, which should not be processed.

Output

For each line, print the actual number being counted, when the m-th person claps for the k-th time. If this can never happen, print ‘`-1`’.

Sample Input

```
4 3 1
4 3 2
4 3 3
4 3 4
0 0 0
```

Sample Output

```
17
21
27
35
```