Code

KEY Inc., the leading company in security hardware, has developed a new kind of safe. To unlock it, you don't need a key but you are required to enter the correct n-digit code on a keypad (as if this were something new!). There are several models available, from toy safes for children (with a 2-digit code) to the military version (with a 6-digit code).

The safe will open as soon as the last digit of the correct code is entered. There is no "enter" key. When you enter more than n digits, only the last n digits are significant. For example (in the 4-digit version), if the correct code is 4567, and you plan to enter the digit sequence 1234567890, the door will open as soon as you press the 7 key.

The software to create this effect is rather simple. In the n-digit version the safe is always in one of 10^{n-1} internal states. The current state of the safe simply represents the last $n-1$ digits that have been entered. One of these states (in the example above, state 456) is marked as the unlocked state. If the safe is in the unlocked state and then the right key (in the example above, 7) is pressed, the door opens. Otherwise the safe shifts to the corresponding new state. For example, if the safe is in state 456 and then you press 8, the safe goes into state 568.

A trivial strategy to open the safe is to enter all possible codes one after the other. In the worst case, however, this will require $n * 10^n$ keystrokes. By choosing a good digit sequence it is possible to open the safe in at most $10^n + n - 1$ keystrokes. All you have to do is to find a digit sequence that contains all n-digit sequences exactly once. KEY Inc. claims that for the military version (n=6) the fastest computers available today would need billions of years to find such a sequence - but apparently they don't know what some programmers are capable of...

Input Specification

The input contains several test cases. Every test case is specified by an integer n. You may assume that $1<=n<=6$. The last test case is followed by a zero.

Output Specification

For each test case specified by n output a line containing a sequence of $10^n + n - 1$ digits that contains each n-digit sequence exactly once.

Sample Input

```
1
2
0
```

Sample Output

```
0123456789
00102030405060708091121314151617181922324252627282933435363738394454647484955657585966768697787988990
```