

Problem F. Quadrat

Input file: `quadrat.in`
Output file: `quadrat.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

It is well-known that for any n there are exactly four n -digit numbers (including ones with leading zeros) that are *self-squares*: the last n digits of the square of such number are equal to the number itself. These four numbers are always suffixes of those four infinite sequences:

```
...0000000000  
...0000000001  
...8212890625  
...1787109376
```

For example, $09376^2 = 87909376$, which ends with 09376.

You need to count the numbers that are *almost self-squares*: such that each of the last n digits of their square is at most d away from the corresponding digit of the number itself. Note that we consider digits 0 and 9 to be adjacent, so for example digits that are at most 3 away from digit 8 are 5, 6, 7, 8, 9, 0 and 1.

Input

The first line of the input file contains the number of testcases t , $1 \leq t \leq 72$. Each of the next t lines contains one testcase: two numbers n ($1 \leq n \leq 18$) and d ($0 \leq d \leq 3$).

Output

For each testcase, output the number of almost self-squares with length n and the (circular) distance in each digit from the square at most d on a line by itself.

Examples

<code>quadrat.in</code>	<code>quadrat.out</code>
2	4
5 0	12
2 1	

Note

In the second example case, the 12 almost self-squares are: 00, 01, 10, 11, 15, 25, 35, 66, 76, 86, 90, 91.