# Find the One for Me

This problem was worth $500$ points and was ranked easy. The author of this problem is Dhruv Chawla.

**Note:** GitHub does not support LaTex in Markdown. If you want a more readable version of the problem, download the PDF file instead.

## Statement

Dhruv really likes the digit $1$. So, Aryan gave Dhruv two numbers $x$ and $y$ for his birthday. The numbers only contain the digit $1$ because otherwise Dhruv will be disappointed.

Dhruv will be happy if Aryan could also answer the question "What is the middle digit in the product $x \cdot y$?".

Since Aryan doesn't know how to solve the problem, he asks for your help. You are required to answer $t$ testcases. Help Aryan make Dhruv happy.

Aryan does not tell you the values of $x$ and $y$ because the numbers could be very large to take as input. He instead gives you integers $k$ and $l$ - the number of digits in $x$ and $y$, respectively.

**Note:** $x$ and $y$ have the same parity, i.e. $x$ and $y$ are either both even or both odd.

## Input Format

The first line of the input contains an integer $t$ - the number of testcases.

Each of the following $t$ lines contain an integer $k$ and $l$ - the number of digits in the numbers $x$ and $y$ that Aryan gave to Dhruv.

## Output Format

The output should contain $t$ lines.

The $i^{th}$ line should contain the answer to the $i^{th}$ testcase - the middle digit in the product $x \cdot y$.

## Constraints

$1 \le t \le 1000$

$1 \le k \le 10^9$

$1 \le l \le 10^9$

Note: $x$ and $y$ only contain $1$ in their digits and no other digit.

## Sample Tests

### Sample Test 1

**Input**

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

**Output**

```
1  1
2  2
```

**Explanation**

In the first test case, $k = 1$ and $l = 3$, so $x = 1$ and $y = 111$.

Their product is $x \cdot y = 111$. The middle digit is $1$.

In the second test case, $k = 2$ and $l = 4$, $x = 11$ and $y = 1111$.

Their product is $x \cdot y = 12221$. The middle digit is $2$.

### Sample Test 2

**Input**

```
1  3
2  999999998 999999998
3  2 1000000000
4  690 42
```

**Output**

```
1  8
2  2
3  6
```

# Solution

**Time Complexity:** $O(t)$

**Space Complexity:** $O(1)$

To solve this problem, we need to take care of a few cases. Problems with a lot of edge case handling is usually discouraged in competitive programming, but they do exist. This problem is good for practice on the same, we think.

Let's take a look at the cases:

- $k = l$: when $x$ and $y$ have the same number of 1's
    - $k \leq 9$: answer is simply $k$

- $k \geq 10$:
  - $k \% 9 = 0$: answer is $9$
  - $k \% 9 = 1$: answer is $0$
  - otherwise answer is $k \% 9 = 1$
- $k \neq l$: when $x$ and $y$ have different number of $1$'s
  - $min(k, l) \% 9 = 0$: answer is $9$
  - otherwise answer is $min(k, l) \% 9$

We leave the proof as an exercise for the readers :)

$$
\begin{array}{r}
111{,}111{,}111 \\
\times\ 111{,}111{,}111 \\
\hline
111{,}111{,}111 \\
1{,}111{,}111{,}110 \\
11{,}111{,}111{,}100 \\
111{,}111{,}111{,}000 \\
1{,}111{,}111{,}110{,}000 \\
11{,}111{,}111{,}100{,}000 \\
111{,}111{,}111{,}000{,}000 \\
1{,}111{,}111{,}110{,}000{,}000 \\
+\ 11{,}111{,}111{,}100{,}000{,}000 \\
\hline
12{,}345{,}678{,}987{,}654{,}321
\end{array}
$$

**Solution in C++:**

```cpp
#include <bits/stdc++.h>

constexpr bool test_cases = true;

void solve () {
   int k, l;
   std::cin >> k >> l;

   if (k == l) {
      if (k <= 9)
         std::cout << k;
      else {
         if (k % 9 == 0)
            std::cout << 9;
         else if (k % 9 == 1)
            std::cout << 0;
         else
```

```cpp
18          std::cout << k % 9;
19      }
20    }
21    else {
22      int m = std::min(k, l);
23      if (m % 9 == 0)
24        std::cout << 9;
25      else
26        std::cout << m % 9;
27    }
28    std::cout << '\n';
29  }
30
31  int main () {
32    std::ios::sync_with_stdio(false);
33    std::cin.tie(nullptr);
34
35    int cases = 1;
36    if (test_cases)
37      std::cin >> cases;
38    while (cases--)
39      solve();
40
41    return 0;
42  }
```