Find the One for Me

This problem was worth 500 points.
The authors of this problem are Aryan V S and 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: k and l have the same parity, i.e. k and l 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

$1 \leq t \leq 1000$

$1 \leq k \leq 10^9$

$1 \leq l \leq 10^9$

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

Sample Tests

Sample Test 1

Input

```
2
1 3
2 4
```

Output

```
1
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

```
3
999999998 999999998
2 1000000000
690 42
```

Output

```
8
2
6
```

Solution

Time Complexity: $O(t)$

Space Complexity: $O(1)$

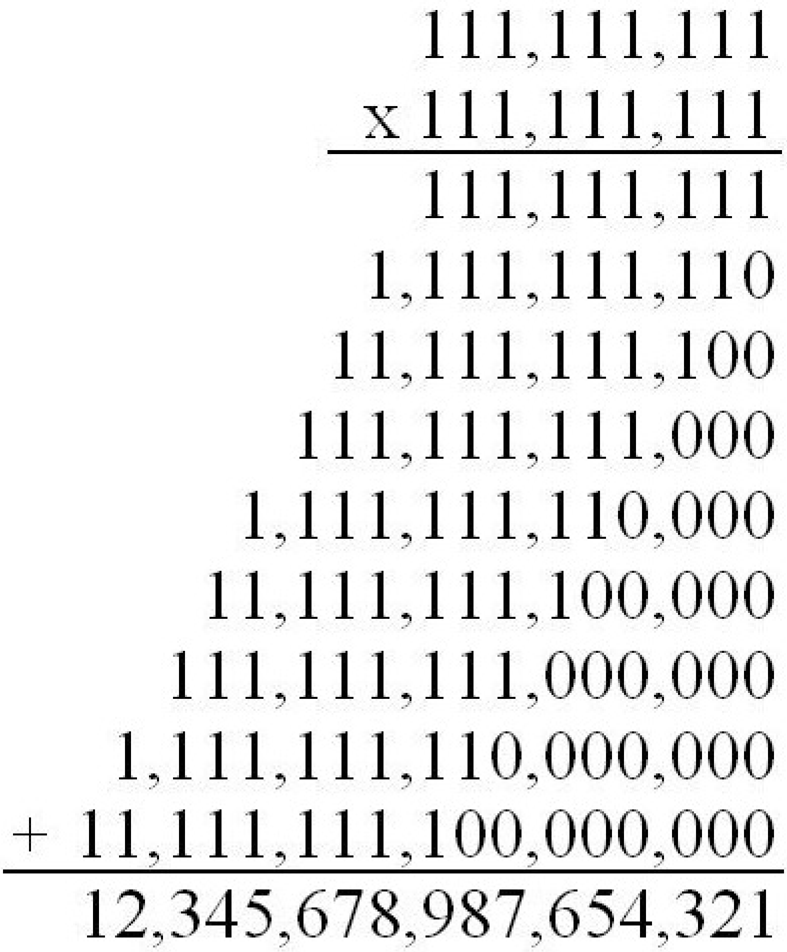
The simple multiplication approach does not work because creating a string of 1s based on the values of k and l will end up taking up too much memory. Also, multiplication is very slow for large numbers.

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 :)



```
                  111,111,111
                x 111,111,111
                -------------
                  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
    -------------------------
   12,345,678,987,654,321
```

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
        std::cout << k % 9;
    }
  }
  else {
    int m = std::min(k, l);
    if (m % 9 == 0)
      std::cout << 9;
    else
      std::cout << m % 9;
  }
  std::cout << '\n';
}

int main () {
  std::ios::sync_with_stdio(false);
  std::cin.tie(nullptr);

  int cases = 1;
  if (test_cases)
    std::cin >> cases;
  while (cases--)
    solve();

  return 0;
}
```