As you stroll by the arena gift shop, you luckily spot another event waiting for you. At the checkout register, the clerk is busy scanning UPC codes. Any product you buy has one of these. It helps identify the product being sold so that the store can manage inventory, pricing and other data. Universal Product Codes (UPC) are easy to spot by their bar code.

problem 2
Check Digit
3 points

0 12345 67890 5

A UPC is a 12 digit number that encodes product information such as the manufacturer, product type, weight, and other data. The last digit in this number is the check digit. This extra digit helps verify that a tired programmer didn't get one digit wrong, or transpose a pair of numbers (ex. 34 -> 43) or otherwise alter the sequence.

Check Digit:
A check digit is used to ensure that a sequence of numbers was transmitted or entered correctly without human error. The algorithm used to calculate the check digit determines the types of errors it will catch. For UPC the algorithm catches 100% of single digit errors and 89% of transposition errors.

Your task is to calculate the missing check digit for the given list of UPCs.

Here's the UPC check digit algorithm:
- First, add all the digits in the odd-numbered positions together and multiply the result by three.
- Then, add the digits in the even-numbered positions to the result.
- Next, find modulo 10 of the sum. Modulo calculates the remainder after dividing the sum by 10.
- Finally, if the remainder is not zero, subtract it from 10.

Input
The first line of input contains the number of UPCs that follow. The digits of each UPC will be separated by one space.

```
6
0 1 2 3 4 5 6 7 8 9 0
0 3 6 0 0 0 2 9 1 4 5
0 7 3 8 5 2 0 0 9 3 8
0 7 0 7 3 4 0 5 3 1 6
0 4 1 2 2 0 1 8 9 0 4
0 3 7 0 0 0 2 0 2 1 4
```

Output
For each UPC, the program must print the UPC including the calculated check digit. The digits of each UPC should be separated by a single space.

```
0 1 2 3 4 5 6 7 8 9 0 5
0 3 6 0 0 0 2 9 1 4 5 2
0 7 3 8 5 2 0 0 9 3 8 5
0 7 0 7 3 4 0 5 3 1 6 0
0 4 1 2 2 0 1 8 9 0 4 5
0 3 7 0 0 0 2 0 2 1 4 1
```