

There are often ways to check (other than brute force division) whether a number is divisible by another number. As for example

- a) In decimal number system a number is divisible by 2 if the rightmost digit is divisible by 2. So **42, 38, 50** are divisible by 2.
- b) In decimal number system a number is divisible by 4 if the number formed by rightmost two digits is divisible by 4. So **100, 124, 1328** are divisible by 4.
- c) In decimal number system a number is divisible by 3 if the sum of its digits is divisible by 3. So 123 is divisible by 3 as $(1+2+3 = 6 = 3*2)$. Similar testing is there for divisibility by 9. For divisibility testing of 99 we need to add up all digits from the right considering two digits as a number. For example **2374310351037933** divisible by 99 as $33 + 79 + 03 + 51 + 03 + 31 + 74 + 23 = 297 = 99 * 3$
- d) A number is divisible by 11 if the difference of the sums of its odd positioned digits and even positioned digits from the right is divisible by 11. As for example **1270401** is divisible by 11, as $(1 - 0 + 4 - 0 + 7 - 2 + 1) = 11$, which is divisible by 11.
- e) A number is divisible by 7 if the difference of the sums of the summation of the numbers formed by alternate three digits from the right is divisible by 7. As for example **1668266767874** divisible by 7 because $(874 - 767 + 266 - 668 + 001) = -294 = -42 * 7$.

In this problem you will have to automatically discover the conditions of divisibility testing for different number systems ($2 < Base < 500$) and different numbers (less than 5000) as divisors. If condition is not found adding all the digits, considering up to 1000 rightmost digits as a number or considering the summation of up to 1000 alternate digits just print the line '**condition not found.**'. If the condition is found then output should be formatted according to the following sample output. When more than one type of condition is found print them according to the order specified above (first try considering the left most digits, then adding all digits, then and then the alternate digit scheme) print that condition and processes the next input. The sample outputs will make everything clear.

Input

First line of the input file contains an integer number N ($N \leq 2000$), which indicates how many sets of input are there. Each of the next N lines makes one set of input. Each line contains two integer numbers B ($2 < B < 500$) and D ($2 \leq D < 5000$). Here B is the base of the number system you have to consider and D is the number for which you have to find the divisibility criteria. Note that you won't have to consider any compound condition. As for example, in decimal number system testing divisibility test for 2 and 3 can test the divisibility by 6. But you should not consider this, as it is a mixture of two rules. You should only consider the individual rules explained at the top.

Output

For each set of input produce one line of output. Output format for all divisibility schemes is shown in the sample output. The sample output contains all the different types of rules. A blank line should be printed between the outputs of two consecutive sets.

Sample Input

```
7
10 3
10 99
10 2
10 4
10 5
10 11
10 7
```

Sample Output

```
Add all 1
Add all 2
Rightmost 1
Rightmost 2
Rightmost 1
Add all 2
Alternate 1 change sign
Add all 6
Alternate 3 change sign
```