Ascending Fibonacci Numbers

John is trying to learn the Fibonacci sequence. This is what he has learned so far. The first two terms of the sequence are f(1) = 0 and f(2) = 1. The next term f(n) is then calculated by adding the previous two terms f(n-1) and f(n-2). Therefore,

f(1) = 0

f(2) = 1

f(3) = f(2) + f(1) = 1 + 0 = 1

f(4) = f(3) + f(2) = 1 + 1 = 2

f(5) = f(4) + f(3) = 2 + 1 = 3

f(6) = f(5) + f(4) = 3 + 2 = 5

After calculating this for a while, John realized that the values are becoming too big. In order to keep the values small, John changed his algorithm. Instead of calculating f(n) = f(n-1)+f(n-2), he decided he will calculate f(n) = (f(n-1)+f(n-2)) % 10^5.

Now John wants to do some research on his new modified Fibonacci sequence. He will give you an integer A (A<=10^5) and an integer B (B<=10^6). You have to output all the terms from f(A) to f(A+B) in ascending order (non-decreasing order). But printing so many numbers is too much of a hassle. So, if there are more than 100 terms in the output, then only print the first 100.

Input

The first line contains an integer **T** (**T**<=100), which is the number of test cases.
Each test case contains two positive integers **A** and **B** as mentioned before.

Output

For each test case, print case number (Check sample output) and then print the terms from f(A) to f(A+B) in ascending order (non-decreasing order). If there are more than 100 terms in the output, then only print the first 100.

Example

Input:

```
3
1 3
3 3
100 1
```

Output:

```
Case 1: 0 1 1 2
Case 2: 1 2 3 5
Case 3: 15075 69026
```