

# The Revenge Of Anti Hash

Given a base **B** and a modulus **M**, the polynomial hash of a string **S**, consisting of only lowercase letters (**a-z**) is defined as below:

```
int GetHashCode(string str, int B, int M){
    long long hash = 0;
    for (auto chr: str)
        hash = (hash * B + chr - 'a' + 1) % M;

    return hash;
}
```

In other words, first the letters of the string are replaced by numbers (equivalent to their position, 'a' gets mapped to 1, 'b' to 2, ... and 'z' to 26). This is then considered to be a number in base **B** (the rightmost number is the least significant digit), and the value of this number taken modulo **M** is called the polynomial hash of the string.

Limak the bear loves to hack other contestants in Codeforces. After the recent educational round, he came to know that his friend Swistak used the polynomial hash function stated above to solve the hardest problem! And believe it or not, he was the only one to solve that problem which eventually made him the round champion! Limak is very angry, how can Swistak solve a problem which Limak himself couldn't solve? And worst of all, Swistak used hashing to solve that problem. Limak believes people who use hashing have no real skill, getting 'Accepted' just implies getting lucky, nothing more.

Limak is just a little bear, he is not very good at solving problems. But after hours of scratching his head, he was able to come up with a solution involving birthday attacks. That should hack Swistak's solution since educational rounds allow hacking for 24 hours after the round ends. And voila! When he coded it later that night, he was finally able to come up with a case that broke Swistak's solution. And down he goes. From the top place to 153, even below Limak! Limak was overwhelmed with joy.

When Swistak woke up the next morning and casually checked the rank list he was furious. He could not believe what he saw, rolling his eyes in disbelief. He was the only contestant to solve the last problem and that earned him the top place. But alas! No more, because someone hacked his solution and he dropped down more than a hundred fifty places. He clicked on the problem to see who hacked him, and his disbelief grew to anger and frustration when he realized it was his friend Limak! He couldn't believe his eyes. "I thought he was my friend, how could he do this to me?", he wondered. He vowed to take revenge. He modified his solution to use double hashing. But just to be extra sure so that Limak can never hack his solution ever again, he hashed it a few more times resulting in **K** total hashes. Then he submitted his solution which passed the tests and Limak's initial hack as expected.

Afterward, he rushed to Limak's place and challenged him to a duel. He claimed Limak was jealous and just got lucky while hacking his solution and has no real hacking skills. Feeling overconfident with his new solution, Swistak challenged Limak to hack his new solution and suggested he will retire from competitive programming if Limak can hack his new solution. But if however, Limak fails, then Limak must retire instead!

Limak, being provoked like this, takes up the challenge without thinking it through. But he has no clue how to solve it, he is just a little bear after all. He thought about it throughout the whole day but has no idea how to crack it. With just 4 hours left before the hacking phase ends, he desperately turns to you for help. He knows this isn't exactly fair, but nothing's fair in love and war as they say and he doesn't want to retire from competitive programming. Not now and not ever. Please help Limak solve the following problem and beat Swistak once and for all, thereby saving his career.

Limak will give you  $K$  pairs of numbers,  $(B_1, M_1), (B_2, M_2), \dots, (B_K, M_K)$ . Each pair consists of a base  $B$  and a modulus  $M$ . These are the numbers Swistak used to hash strings  $K$  times in his new solution. Limak needs you to find **two different strings** consisting of lowercase letters only. The strings must have the same hash value when hashed with each of the  $K$  base/mod pairs with the above described function. Since Codeforces will not accept just any string of arbitrary length as hack inputs, each of the strings also need to be non-empty and cannot exceed more than **65536** characters in length. They can be of different lengths though.

## Input

The first line contains  $T$ , denoting the number of test cases. Then  $T$  test cases follow. The first line of each case contains an integer  $K$ . The next  $K$  lines consist of two integers  $(B_i, M_i)$ . These are the base and mod pairs Swistak used in his hash function.

## Constraints

- $1 \leq T \leq 20$
- $1 \leq K \leq 12$
- $32 \leq B_i \leq 256$
- $1 \leq M_i \leq 256$

## Output

For each test case, output the required two strings separated by a single space. If there is more than one solution satisfying all the above criteria then you may output any of them. You can be assured that there will always be at least one pair of strings.

## Sample Input

```
1
1
32 1
```

## Sample Output

```
hello world
```

## Challenge

You might also enjoy:

[Anti Hash](#)

[Anti Hash II](#)