

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 Hash(string S, int B, int M){
    long long H = 0;
    for (int i = 0; i < S.length(); i++){
        H = (H * B + S[i] - 'a' + 1) % M;
    }
    return H;
}
```

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** (rightmost number is the least significant digit), and the value of this number in **base 10** 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! Limak is so 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 uses hashing have no real skill, getting *Accepted* just implies getting lucky, nothing more!

Later that night, Limak realized that he can hack the solution if he is able to solve the following problem efficiently. Limak felt triumphant, he will teach Swistak and that stupid hash function of his a lesson! But Limak is just a little bear, he is not very good at solving problems. Please help Limak solve the following problem so that he can hack Swistak's solution.

Limak will give you a string **S** of length **N**, consisting of only lowercase letters, a base **B** and a modulus **M**. Your task is to find another string **T**, satisfying all of the following constraints:

- Length of **T** is exactly **N**
- **T** consists of only lowercase letters (**a-z**)
- **T** and **S** are two different strings
- **T** and **S** have the same hash, i.e. $\text{Hash}(\mathbf{S}, \mathbf{B}, \mathbf{M}) = \text{Hash}(\mathbf{T}, \mathbf{B}, \mathbf{M})$

Input

The first line contains **Q**, denoting the number of test cases. Each test case consists of two lines. The first line of each case contains three integers, **N**, **B**, **M**. The next line contains the string **S** of length **N**, consisting of only lowercase letters.

Constraints

- $1 \leq Q \leq 30$
- $10^5 \leq N \leq 10^6$
- $10^5 \leq B < 2^{31}$
- $10^5 \leq M < 2^{31}$
- $S_i \in \{a-z\}$
- $B \neq M$ and both B and M are prime numbers

Output

For each test case, output the string T in a single line. It is guaranteed that such a string will always exist for the given constraints. If there are many solutions, you can output any of them.

Sample Input

```
1
38 666666667 1000000009
bbababbbbbbaabaababaabbababbababababb
```

Sample Output

```
hisotomeseemslikeanotoriouscoincidence
```

Note

The sample input contains a string of length 38 only for demonstration and clarity. There will be no such cases in the judge data, every case will strictly satisfy the constraints mentioned above.

Challenge

You might also enjoy:

[Anti Hash II](#)

[The Revenge Of Anti Hash](#)