ACM-ICPC Live Archive

7258 Frog and String

Frog studies algorithms on strings. He finds it so interesting that he can't stop playing with his strings. These days he has just learnt about palindrome, and comes up with a problem about it.

Given two integers, N and M, he wants to construct a string of length N, whose substrings contain exactly M distinct non-empty palindromes. A palindrome is a string which is exactly the same as the reverse of itself. For example, "ABBA", "ADA", "A", and "UUSSUU" are palindromes, but "USTC", "AB", and "ABC" are not. A substring is a consecutive part of the original string. For example, "US", "USTC", "STC", and "TC" are substrings of "USTC", but "UC" and "CT" are not.

Frog finds it too hard for him to solve this problem. So he asks you for help. BTW, he won't make it too easy for you, so he decided to ask you solve this problem under his restrictions. You can only use the first K capital letters in the English alphabet (`A-Z`). Please write a program to solve this problem.

Input

There is an integer T in the first line indicating the number of total testcases ($T \leq 20000$). Each test case contains three integers N, M, and K, ($1 \leq N, M \leq 100000$, $1 \leq K \leq 26$), separated by single spaces. We guarantee the sum of N will not exceed 2000000.

Output

For each test case, output a single line consisting of '`Case #`X`:`' first, where X is the test case number starting from 1. Output the string that you find in the next line. The string should contain only the first K capital letters. If there are multiple solutions, you can output any of them. If there is no such string satisfying Frog's requirements, output '`Impossible`' instead. Please follow the output format exactly, and do not output any additional character or new line.

Hint:

For the first test case, "A", "ABA", "B" are the all distinct palindrome substrings of "ABA". There are other possible answers, such as "BAB" and "AAA". For the second test case, "USTC" is not a valid answer, because it contains letters other than the first 4 capital letters.

Sample Input

```
4
3 3 3
4 4 4
2 2 1
2 1 1
```

Sample Output

```
Case #1:
ABA
Case #2:
ABCD
Case #3:
AA
Case #4:
```

```
Impossible
```