Rotate String

Scturtle likes strings very much. He is getting bored today, because he has already completed this week's task and doesn't have anything else to do. So he starts left-rotating a string. If the length of the string is n, then he will rotate it n times and note down the result of each rotation on a paper.

For a string $S = s_1 s_2 \dots s_n$, n rotations are possible. Let's represent these rotations by $r_1, r_2 \dots r_n$. Rotating it once will result in string $r_1 = s_2 s_3 \dots s_n s_1$, rotating it again will result in string $r_2 = s_3 s_4 \dots s_n s_1 s_2$ and so on. Formally, i^{th} rotation will be equal to $r_i = s_{i+1} \dots s_{n-1} s_n s_1 \dots s_i$. Note that $r_n = S$.

Your task is to display all n rotations of string S.

For example, if $S =$ `abc` then it has 3 rotations. They are $r_1 =$ `bca`, $r_2 =$ `cab` and $r_3 =$ `abc`.

Input Format

The first line contains an integer, T, which represents the number of test cases to follow. Then follows T lines, which represent a test case each.
Each test case contains a string, S, which consists of lower case latin characters $('a' - 'z')$ only.

Output Format

For each test case, print all the rotations, $r_1\ r_2 \dots r_n$, separated by a space.

Constraints

$1 \leq T \leq 10$
$1 \leq n \leq 10^2$
S will consist of lower case latin character, $['a' \dots 'z']$ only.

Sample Input

```
5
abc
abcde
abab
aaa
z
```

Sample Output

```
bca cab abc
bcdea cdeab deabc eabcd abcde
baba abab baba abab
aaa aaa aaa
z
```

Explanation

Test case #1: This case is mentioned in the problem statment.
Test case #2: Rotations of `abcde` are: `bcdea` -> `cdeab` -> `deabc` -> `eabcd` -> `abcde`.
Test case #3: Rotations of `abab` are: `baba` -> `abab` -> `baba` -> `abab`.
Test case #4: All three rotations will result into same string.
Test case #5: Only one rotation is possible, and that will result into original string.

Tested by: Lalit Kundu