Permutations

Time Limit: 1.0s **Memory Limit:** 32M

In this programming problem, you are required to write a function named `permute` that return a list which contains all permutation of the given integer list. In other words, given a list of **distinct sorted** integers, return a list contains all the possible permutations in **ascending** order of list.

You can start with this template code and ensure your code can run correctly within this template:

```
def permute(nums):
    # your code here

# Note that the following code is for local testing purposes only. You should leave
this part of code unchanged and not submit it to the OJ system.
T = int(input())
for t in range(T):
    nums = list(map(int, input().split()))
    print(permute(nums))
```

Please note that you only have to submit the definition of the `permute` function to the OJ system. The rest of the code is for your local testing purposes only.

Input Specification

The first line of the input gives the number of test cases, , and then test cases follow.

For each test case, it will consist of a single line that contains integers separated by spaces.

Constraints

Output Specification

For each test case, output a single line which contains the list of all possible permutations of the given integer list. Note that the permutations should be in ascending order of list.

Sample Input

```
2
0 1
1 2 3
```

Sample Output

```
[[0, 1], [1, 0]]
[[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]
```

Hint

`recursion` `for-loop`