

Permutations of a given string

Given a string S . The task is to print all permutations of a given string.

Input:

The first line of input contains an integer T , denoting the number of test cases. Each test case contains a single string S in capital letter.

Output:

For each test case, print all permutations of a given string S with single space and all permutations should be in lexicographically increasing order.

Constraints:

$$1 \leq T \leq 10$$

$$1 \leq \text{size of string} \leq 5$$

Example:

Input:

2

ABC

ABSG

Output:

ABC ACB BAC BCA CAB CBA

ABGS ABSG AGBS AGSB ASBG ASGB BAGS BASG BGAS BGSA BSAG BSGA
GABS GASB GBAS GBSA GSAB GSBA SABG SAGB SBAG SBGA SGAB SGBA

Explanation:

Testcase 1: Given string ABC has permutations in 6 forms as ABC, ACB, BAC, BCA, CAB and CBA .

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. void perm(string ip,string op)
4. {
5.     if(ip.length()==0)
6.     {
7.         cout<<op<<" ";
8.         return;
9.     }
10.    for(int i=0;i<ip.length();i++)
11.    {
12.        string ip1=ip;
13.        string op1=op;
14.        op1=op1+ip[i];
15.        ip1.erase(ip1.begin()+i);
16.        perm(ip1,op1);
17.    }
18.}
19.void solve()
20.{
21.string ip;
22.cin>>ip;
23.sort(ip.begin(),ip.end());
24.string op="";
25.perm(ip,op);
26.}
27.int main()
28.{
29.    int t;
```

```
30. cin>>t;  
31. while(t--)  
32. {  
33.     solve();  
34.     cout<<endl;  
35. }  
36.}
```

