

## 1199: 全排列

### 题目描述

给定一个由不同的小写字母组成的字符串，输出这个字符串的所有全排列。

我们假设对于小写字母有 ‘a’ < ‘b’ < ... < ‘y’ < ‘z’，而且给定的字符串中的字母已经按照从小到大的顺序排列。

### 输入格式

只有一行，是一个由不同的小写字母组成的字符串，已知字符串的长度在1到6之间。

### 输出格式

输出这个字符串的所有排列方式，每行一个排列。要求字母序比较小的排列在前面。字母序如下定义：

已知 $S=s_1s_2\dots s_k$ ,  $T=t_1t_2\dots t_k$ ，则 $S<T$ 等价于，存在 $p(1\leq p\leq k)$ ，使得 $s_1=t_1, s_2=t_2, \dots, s_{p-1}=t_{p-1}, s_p<t_p$ 成立。

### 输入样例

abc

### 输出样例

abc  
acb  
bac  
bca  
cab  
cba

## 解析

参考1317题目

## 编码

```
#include<iostream>

using namespace std;

const int N = 1e5 + 10;

//原始字符串
char a[N];
//组合的新字符串
char b[N];

int len, n;

//标记当前字母是否被访问过
bool vis[N];

//输出拼组的字符
void print() {
    for (int i = 0; i < n; i++) {
        cout << b[i];
    }
    cout << "\n";
}

//执行深搜算法
void dfs(int x) {
    if (x > len) {
        print();
        return;
    }
    for (int i = 0; i < len; i++) {
        if (vis[i]) {
            continue;
        }
        //拼组新的字符
        b[n++] = a[i];
        //将当前使用的字符进行标记,防止被二次使用
        vis[i] = true;
        //继续拼组下一个字符
        dfs(x + 1);
        //回溯
        vis[i] = false;
        n--;
    }
}
```

```
}  
  
int main(int argc, char **argv) {  
    cin >> a;  
    len = strlen(a);  
    dfs(1);  
    return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

