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。

**Input**

输入只有一行，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。

**Output**

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...s k , T = t 1t 2...t k，则S < T 等价于，存在p (1 <= p <= k)，使得   
s 1 = t 1, s 2 = t 2, ..., s p - 1 = t p - 1, s p < t p成立。

**Sample Input**

abc

**Sample Output**

abc

acb

bac

bca

cab

cba

破编译系统还过不了c++的

#include<stdio.h>

//#include<iostream>

#include<string.h>

using namespace std;

char s[10],a[10];

int l;

int b[10];

void dfs(int x)//搜第几个数

{

if(x==l)

{

for(int i=0;i<l;i++)printf("%c",a[i]);//cout<<a[i];

printf("\n");

return;

}

for(int i=0;i<l;i++)//枚举所有的字母选择

{

if(b[i]==0)

{

a[x]=s[i];

b[i]=1;

dfs(x+1);

b[i]=0;

}

}

}

int main()

{

// freopen("input.txt","r",stdin);

gets(s);

l=strlen(s);

memset(b,0,sizeof(b));

dfs(0);

return 0;

}

并行着的两个循环，一个搜第几个数（第几位数），令一个搜所以的字母选择（即在这个位上该放哪个数）

这两个循环一个是由dfs进行的，一个是dfs中的for循环进行的

再开一个数组判断某字母是否被选择

附

[OpenJ\_Bailian - 4070](https://cn.vjudge.net/problem/614087/origin) 全排列

#include<bits/stdc++.h>

using namespace std;

int n;

int a[10];

int b[10];

void dfs(int k)

{

if(k==n+1)

{

cout<<a[1];

for(int i=2;i<=n;i++)cout<<' '<<a[i];

cout<<endl;

return;

}

for(int i=1;i<=n;i++)

if(b[i]==0)

{

a[k]=i;

b[i]=1;

dfs(k+1);

b[i]=0;

}

}

int main()

{

while(cin>>n&&n)

{

memset(a,0,sizeof(a));

memset(b,0,sizeof(b));

dfs(1);

}

}