解题报告

参考答案

```
#include<iostream>
#define N 20
using namespace std;
int n,p[N];
bool vst[N];
void print(){
    for(int i=0;i<n-1;i++)cout<<p[i]<<" ";
    cout<<p[n-1]<<endl;
}
void dfs(int x){
    if(x==n){print();return;}
    for(int i=1;i<=n;i++)
        if(!vst[i]){
            vst[i]=1;p[x]=i;
            dfs(x+1);
            vst[i]=0;
        }
}
int main(){
    cin>>n;
    dfs(0);
    return 0;
}
```

题目分析

题目考察枚举全排列的理解和运用，需要注意的是：和课件例题不同，需要对全部 n 个数全排列，而不是只排列前 m 个。

提示

题目是课件例题“前 m 名”的变化，可参考课件第 9 页的程序。
因为 n 不固定，所以不能使用 n 层循环的程序结构。要使用 dfs 框架来实现 n 层枚举。利用 p 数组记录每步所选择的数字。
因为每个数字只能用一次，所以需要 vst 数组来记录某个数字是否已经用过。

易错点

1） 受例题“一维染色”影响，没有使用 vst 数组。
2） dfs(x)中递归调用完 dfs(x+1)后，没有把 vst[i]恢复为 0。

解题报告

参考答案

```
#include<iostream>
#include<string>
#include<algorithm>
#define N 10
using namespace std;
string names[N];
int n,p[N];
bool vst[N];
void print(){
    for(int i=0;i<n;i++)cout<<names[p[i]];
    cout<<" wang"<<endl;
}
void dfs(int x){
    if(x==n){print();return;}
    for(int i=0;i<n;i++)
        if(!vst[i]){
            vst[i]=1;p[x]=i;
            dfs(x+1);
            vst[i]=0;
        }
}
int main(){
    cin>>n;
    for(int i=0;i<n;i++)cin>>names[i];
    sort(names,names+n);
    dfs(0);
    return 0;
}
```

题目分析

题目考察枚举全排列的理解和运用，需要注意的是：和课件例题不同，需要对全部 n 个音节全排列，而不是只排列前 m 个。

提示

题目是课件例题“前 m 名”的变化，可参考课件第 9 页的程序。

因为 n 不固定，所以不能使用 n 层循环的程序结构。要使用 dfs 框架来实现 n 层枚举。利用 p 数组记录每步所选择的音节。

因为每个音节只能用一次，所以需要 vst 数组来记录某个数字是否已经用过。

易错点

1） 读入的 n 个音节没有排序，输出的排列组合不是按字典序。

2） dfs(x)中递归调用完 dfs(x+1)后，没有把 vst[i]恢复为 0。

3） pirnt 函数打印完一种排列后，忘记输出“ wang”。

解题报告

参考答案

```
#include<bits/stdc++.h>
#define N 15
#define nCOLORS 3
using namespace std;
string s;
char colors[nCOLORS]={'B','G','R'};
int n,p[N];
void print(){
    for(int i=0;i<n;i++)cout<<colors[p[i]];
    cout<<endl;
}
void dfs(int x){
    if(x==n){print();return;}
    if(p[x]>=0){dfs(x+1);return;}
    for(int i=0;i<nCOLORS;i++)
        if(x==0&&p[1]!=i||x==n-1&&p[n-2]!=i||x>0&&x<n-1&&p[x-
1]!=i&&p[x+1]!=i){
            p[x]=i;
            dfs(x+1);
            p[x]=-1;
        }
}
int main(){
    freopen("color.in","r",stdin);
    freopen("color.out","w",stdout);
    cin>>s;
    n=s.size();
    fill(p,p+n+1,-1);
    for(int i=0;i<n;i++)
        if(s[i]=='B')p[i]=0;
        else if(s[i]=='G')p[i]=1;
        else if(s[i]=='R')p[i]=2;
    dfs(0);
    return 0;
}
```

题目分析

题目考察枚举全排列的理解和运用，需要注意的是：此题不是“一维染色”，而是“一维染色有定色”。

提示

题目就是课件例题“一维染色有定色”，可参考课件第 19 页的代码填空版本，补充完整。因为格子数量 n 不固定，所以不能使用 n 层循环的程序结构。要使用 dfs 框架来实现 n 层枚举。

利用 p 数组记录每个格子的颜色编号（0～2 分别代表 BGR），初始没有定色的格子设置为-1。 dfs(x)函数考察第 x 格子。如果其值不是-1，说明有定色，跳过这个格子，直接递归调用 dfs(x+1)，考察下一格子。如果没有定色，枚举各种颜色 i 试图填在格子里，只有相邻格子不同色，才可以填入，递归调用 dfs(x+1)考察下一格子。
易错点
1） 没有对 p 数组赋初值-1。 2） 考察相邻格子是否同色时，没有对开头结尾格子进行特判，造成数组越界。 3） dfs(x)中递归调用完 dfs(x+1)后，没有把 p[x]恢复为-1。

解题报告

参考答案

```
#include<bits/stdc++.h>
#define N 15
#define nCOLORS 4
using namespace std;
string s;
char colors[nCOLORS]={'A','B','C','D'};
int n,p[N];
void print(){
    for(int i=1;i<=n;i++)cout<<colors[p[i]];
    cout<<endl;
}
void dfs(int x){
    if(x==n+1){print();return;}
    if(p[x]>=0){dfs(x+1);return;}
    for(int i=0;i<nCOLORS;i++)
        if(p[x-1]!=i&&p[x+1]!=i){
            p[x]=i;
            dfs(x+1);
            p[x]=-1;
        }
}
int main(){
    freopen("color2.in","r",stdin);
    freopen("color2.out","w",stdout);
    cin>>s;
    n=s.size();
    fill(p,p+n+2,-1);
    for(int i=0;i<n;i++)
        if(s[i]=='A')p[i+1]=0;
        else if(s[i]=='B')p[i+1]=1;
        else if(s[i]=='C')p[i+1]=2;
        else if(s[i]=='D')p[i+1]=3;
    dfs(1);
    return 0;
}
```

题目分析

题目考察枚举全排列的理解和运用，需要注意的是：

1. 此题在 732“格子染色 1”的基础上，增加了一种颜色。
2. 为了在判断相邻格子是否同色时不用对首尾格子特判（以免数组越界），p 数组从 1 号位置开始记录信息，故意浪费了 0 号位置。

提示

题目就是课件例题“一维染色有定色”的变形，可参考课件第 19 页的代码填空版本，补充

完整。
因为格子数量 n 不固定，所以不能使用 n 层循环的程序结构。要使用 dfs 框架来实现 n 层枚举。
利用 p 数组记录每个格子的颜色编号（0～3 分别代表 ABCD），初始没有定色的格子设置为-1。
dfs(x)函数考察第 x 格子。如果其值不是-1，说明有定色，跳过这个格子，直接递归调用 dfs(x+1)，考察下一格子。如果没有定色，枚举各种颜色 i 试图填在格子里，只有相邻格子不同色，才可以填入，递归调用 dfs(x+1)考察下一格子。

易错点

1） 没有对 p 数组赋初值-1。
2） dfs(x)中递归调用完 dfs(x+1)后，没有把 p[x]恢复为-1。