1 根据题目我们可以知道总共有34种牌，分别是（9张饼+9张条+9张万+东南西北+中发白）
2 题目明确说明“胡牌”的请况是“将+刻子(>=0)+顺子(>=0)”，那么我们知道最多有34总牌，那么我们只要去枚举每一种是否可以作为将，然后去判断剩下的是否满足刻子和顺子即可
3 注意题目明确说明如果输入的时候是4张一样的牌，那么这个牌是不可能听的。

代码:

[cpp] view plain copy

```cpp
#include<cstdio>
#include<cstring>
#include<iostream>
#include<algorithm>
using namespace std;

const int N = 34;
const char majong[N][N] = {
"1T","2T","3T","4T","5T","6T","7T","8T","9T",
"1S","2S","3S","4S","5S","6S","7S","8S","9S",
"1W","2W","3W","4W","5W","6W","7W","8W","9W",
"DONG","NAN","XI","BEI","ZHONG","FA","BAI"
};
int ma[N];//记录每一种牌的个数

//返回拍的编号
int init(char *s){
    for(int i = 0 ; i < 34 ; i++)
       if(!strcmp(majong[i] , s))
          return i;
}

//判断当前将是否满足胡牌
bool isOk2(int pos){
    //枚举刻子
    for(int i = 0 ;  i< 34 ; i++){
       if(ma[i] >= 3){
          if(pos == 3)//为什么是3退出呢，因为最多4个刻子或者顺子，那么这里由于ma[i]>=3占了一个那么dep=3加起来就是4个了，所以dep为3返回
             return true;
          ma[i] -= 3;
          if(isOk2(pos+1))//递归回来判断
             return true;
          ma[i] += 3;
       }
    }
```

```
36      //枚举顺子(因为东南西北和中发白是不可能构成顺子的)
37      for(int i = 0 ; i <= 24 ; i++){
38          if(i%9 <= 6 && ma[i] >= 1 && ma[i+1] >= 1 && ma[i
    +2] >= 1){
39              if(pos == 3)//为什么是3退出呢，因为最多4个顺子或者顺子，那么
    这里由于ma[i]>=3占了一个那么dep=3加起来就是4个了，所以dep为3返回
40                  return true;
41              ma[i]--;
42              ma[i+1]--;
43              ma[i+2]--;
44              if(isOk2(pos+1))//递归回来判断
45                  return true;
46              ma[i]++;
47              ma[i+1]++;
48              ma[i+2]++;
49          }
50      }
51      return false;
52  }
53
54  //枚举选将的所有可能
55  bool isOk(){
56      for(int i = 0 ; i < 34 ; i++){//枚举将牌
57          if(ma[i] >= 2){
58              ma[i] -= 2;
59              if(isOk2(0))
60                  return true;
61              ma[i] += 2;//回溯
62          }
63      }
64      return false;
65  }
66
67  int main(){
68      int Case = 1;
69      int tmp[N];
70      char ch[N];
71      bool mark;
72      while(scanf("%s" , ch)){
73          if(ch[0] == '0')
74              break;
75          //读入
76          tmp[0] = init(ch);
77          for(int i = 1 ; i < 13 ; i++){
78              scanf("%s" , ch);
79              tmp[i] = init(ch);
80          }
81          mark = false;
82          printf("Case %d:" , Case++);
83          //暴力枚举34张牌可能的情况
84          for(int i = 0 ; i < 34 ; i++){
```

```
85              memset(ma , 0 , sizeof(ma));//每一次初始化为全0
86              for(int j = 0 ; j < 13 ; j++)
87                  ma[tmp[j]]++;
88              if(ma[i] == 4)//如果牌是四张那么是不可能听的
89                  continue;
90              ma[i]++;//假设拥有这一张牌
91              if(isOk()){
92                  mark = true;
93                  printf(" %s" , majong[i]);
94              }
95          }
96          if(!mark)//如果所有的牌都没有听
97            printf(" Not ready");
98          printf("\n");
99      }
100     return 0;
101 }
```