

斗地主之顺子	题目描述：
在斗地主扑克牌游戏中， 扑克牌由小到大的顺序为：3,4,5,6,7,8,9,10,J,Q,K,A,2，玩	
家可以出的扑克牌阵型有：单张、对子、顺子、飞机、炸弹等。	
其中顺子的出牌规则为：由至少 5 张由小到大连续递增的扑克牌组成，且不能包含 2。	
例如：{3,4,5,6,7}、{3,4,5,6,7,8,9,10,J,Q,K,A}都是有效的顺子；而{J,Q,K,A,2}、	
{2,3,4,5,6}、{3,4,5,6}、{3,4,5,6,8}等都不是顺子。	
给定一个包含 13 张牌的数组，如果有满足出牌规则的顺子，请输出顺子。	
如果存在多个顺子，请每行输出一个顺子，且需要按顺子的第一张牌的大小（必须从小到大）依次输出。	
如果没有满足出牌规则的顺子，请输出 No。	
输入描述：	
13 张任意顺序的扑克牌，每张扑克牌数字用空格隔开，每张扑克牌的数字都是合法的，并且不包括大小王：	
2 9 J 2 3 4 K A 7 9 A 5 6	
不需要考虑输入为异常字符的情况	
输出描述：	
组成的顺子，每张扑克牌数字用空格隔开：	
3 4 5 6 7	
补充说明：	
示例 1	
输入：	

2 9 J 2 3 4 K A 7 9 A 5 6

输出：

3 4 5 6 7

说明：

13 张牌中，可以组成的顺子只有 1 组：3 4 5 6 7

示例 2

输入：

2 9 J 10 3 4 K A 7 Q A 5 6

输出：

3 4 5 6 7

9 10 J Q K A

说明：

13 张牌中，可以组成 2 组顺子，从小到大分别为：3 4 5 6 7 和 9 10 J Q K A

示例 3

输入：

2 9 9 9 3 4 K A 10 Q A 5 6

输出：

No

说明：

13 张牌中，无法组成顺子

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    int n = 13;
    char ch ;
```

```

const int cMaxCard = 16;
int cards[cMaxCard] = {0}; //表示 0 - 10, J Q K A, 空牌 (哨兵)
string strCards[cMaxCard] = {"", "", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A", ""};
//获取输入
while(n--)
{
    cin >> ch;
    if((ch >= '2') && (ch <= '9'))
    {
        cards[ch-'0']++;
    }
    else if(ch == '1') //输入了 10
    {
        cards[10]++;
        cin >> ch;
    }
    else if(ch == 'J'){
        cards[11]++;
    }
    else if(ch == 'Q'){
        cards[12]++;
    }
    else if(ch == 'K'){
        cards[13]++;
    }
    else if(ch == 'A'){
        cards[14]++;
    }
}
int iContinue = 0;
int iStart = 0;
bool isHave = false;
//最多两对顺子
for(int jj = 0; jj < 2; jj++)
{
    iContinue = 0;
    for(int i = 3; i < cMaxCard; i++)
    {
        if(cards[i])
        {

```

```

        iContinue++;
        if(iContinue == 5)
        {
            iStart = i - 4;
        }
    }
    else {
        if(iContinue >= 5) //输出顺子
        {
            isHave = true;
            for(int j = iStart; j < (iStart + iContinue - 1); j++)
            {
                cout << strCards[j] << " ";
                cards[j]--;
            }
            cout << strCards[iStart+iContinue-1] << endl;
            cards[iStart+iContinue-1]--;
        }
        iContinue = 0;
    }
}

if(!isHave) cout << "No";

}

// 64 位输出请用 printf("%lld")

```