

## C++方向编程题答案

### 答案说明：

大家如果对本次题目或者答案有问题，可以联系下方的出题老师答疑。

### 出题老师：

选择题：鲍松山 qq：365690203

代码题：鲍松山 qq：365690203

## 第四周

### day21

#### 题目ID：46126-洗牌

链接：<https://www.nowcoder.com/practice/5a0a2c7e431e4fbbbb1ff32ac6e8dfa0?tpId=85&&tqId=29848&rp=1&ru=/activity/oj&ru=/ta/2017test/question-ranking>

#### 【题目解析】：

本题题目的意思是模拟洗牌过程，牌被分成两组，且每组数量相等，然后每组牌从最后一个牌交叉排列，最后再把排列逆序就是一次洗牌的过程，K次需要重复K次这样的过程。

#### 【解题思路】：

每次读取一个数之后，算出他经过k次洗牌后的位置，只用一个长度为2n数组用来输出

根据当前数的位置，可以算出经过一次洗牌后的位置

如果当前数小于等于n（即在左手），则他下次出现的位置是 2\*当前位置

与之对应的当前位置 + n（即在右手）的牌,则他下次出现的位置是 2\*当前位置 + 1

#### 【示例代码】

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

int main()
{
    int T, n, k;
    cin >> T;
    while(T-->
    {
        cin >> n >> k;
        int num = 2 * n;
        vector<int> card(num);
        for(int i=0; i<num; ++i)
            cin >> card[i];

        //开始洗牌
```

```

for(int i=0; i<k; ++i)
{
    vector<int> tmp(card.begin(), card.end());
    for(int j=0; j<n; ++j)
    {
        card[2*j] = tmp[j];    //左手的牌排放的位置
        card[2*j+1] = tmp[j+n]; //右手的牌排放的位置
    }
}

//输出洗牌的顺序
for(int i=0; i<num-1; ++i)
    cout<<card[i]<<" ";
cout<<card[num-1]<<endl; //最后一张牌后面不能有空格
}
return 0;
}

```

### 题目ID:36888-MP3光标位置

链接: <https://www.nowcoder.com/practice/eaf5b886bd6645dd9cfb5406f3753e15?tpId=37& tqId=21287&rp=1&ru=/activity/oj&qu=/ta/huawei/question-ranking>

#### 【题目解析】：

本题的意思是第一行输入歌曲数量，第二行输入指令，最后需要显式的输出也为两行，第一行为当前歌曲所在的列表，第二行为光标所指向的歌曲。

#### 【解题思路】：

本题比较简单，通过解析指令，进行移动即可，分两种情况，歌曲数目不大于4和大于4的情况。

#### 【示例代码】

```

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

int main()
{
    int n;
    string cmd;
    while(cin >> n >> cmd)
    {
        //将n首歌进行编号1: n,其中num代表当前光标所在的歌曲编号，first代表当前页的第一首歌曲的编号
        int num = 1, first = 1;
        if(n <= 4)
        {
            //歌曲总数<=4
            for(int i=0; i<cmd.size(); ++i)
            {
                //解析命令
                if(num==1 && cmd[i]=='U')

```

```

        num = n;
    else if(num==n && cmd[i]=='D')
        num = 1;
    else if(cmd[i] == 'U')
        num--;
    else
        num++;
}
for(int i=1; i<=n; ++i)
    cout<<i<<" ";
cout<<endl;
cout<<num<<endl;
}
else
{
    //歌曲总数>4
    for(int i=0; i<cmd.size(); ++i)
    {
        //解析命令
        if(first==1 && num==1 && cmd[i]=='U')
        {
            first = n-3; //将first跳入最后一页
            num = n;
        }
        else if(first==n-3 && num==n && cmd[i]=='D')
        {
            first = num = 1;
        }
        else if(first!=1 && num==first && cmd[i]=='U')
        {
            first--;
            num--;
        }
        else if(first!=n-3 && num==first+3 && cmd[i]=='D')
        {
            first++;
            num++;
        }
        else if(cmd[i] == 'U')
            num--;
        else
            num++;
    }
    for(int i=first; i<=first+3; ++i)
        cout<<i<<" ";
    cout<<endl;
    cout<<num<<endl;
}
}
return 0;
}

```