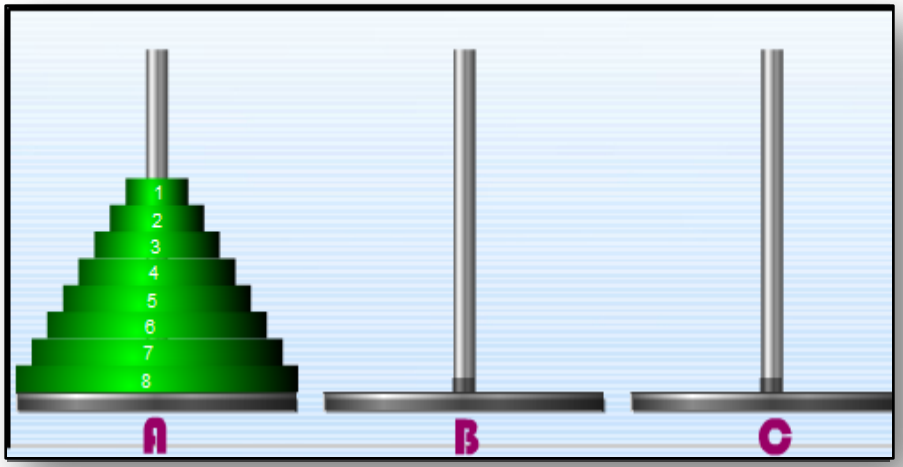
Hanoi塔问题

杨振平

例如：Hanoi塔问题。

- 有A,B,C三根柱子，在A柱子上有n个大小不同的金盘，大盘在下，小盘在上。
- 要将A柱子上的金盘移动到C柱子上，每次只能搬动一个金盘，搬动过程中可以借助任何一根柱子暂时存放金盘，但必须满足大盘在下，小盘在上的条件。

编程显示盘子移动的过程。n由用户输入。



算法分析：

- 如果只有一个盘子,只需一步,直接从A柱移动到C柱, 用A->C表示；
- 如果有2个盘子,共需要移动3步:

  （1） 把A柱上的小盘子移动到B柱;用A->B表示；
  （2） 把A柱上的大盘子移动到C柱；用A->C表示；
  （3） 把B柱上的小盘子移动到C柱；用B->C表示；

- 如果要将A柱上的n个盘子（n值较大），移动到C柱上去，必须先把上面的n-1个盘子从A柱移动到B柱上暂存,按这种思路,就可以把n个盘子的移动过程分作三大步：

  （1） 把A柱上面的n-1个盘子移动到B柱；
  （2） 把A柱上剩下的一个盘子移动到C柱；
  （3） 把B柱上面的n-1个盘子移动到C柱；

演示

- 其中n-1个盘子的移动过程又可按同样的方法分为三大步,这样就把移动过程转化为一个递归的过程,直到最后只剩下一个盘子,按照移动一个盘子的方法移动,递归结束。

【算法描述】n个金盘汉诺塔问题。

```
hanoi(n,A,B,C)
  如果n=1,则
    显示A--->C  //将一个金盘直接从A移到C上
  否则
    hanoi(n-1,A,C,B)  //将n-1个金盘借助C从A移到B上
    显示A--->C
    hanoi(n-1,B,A, C)  //将n-1个金盘借助A从B移到C上
```

汉诺塔问题程序代码-move函数

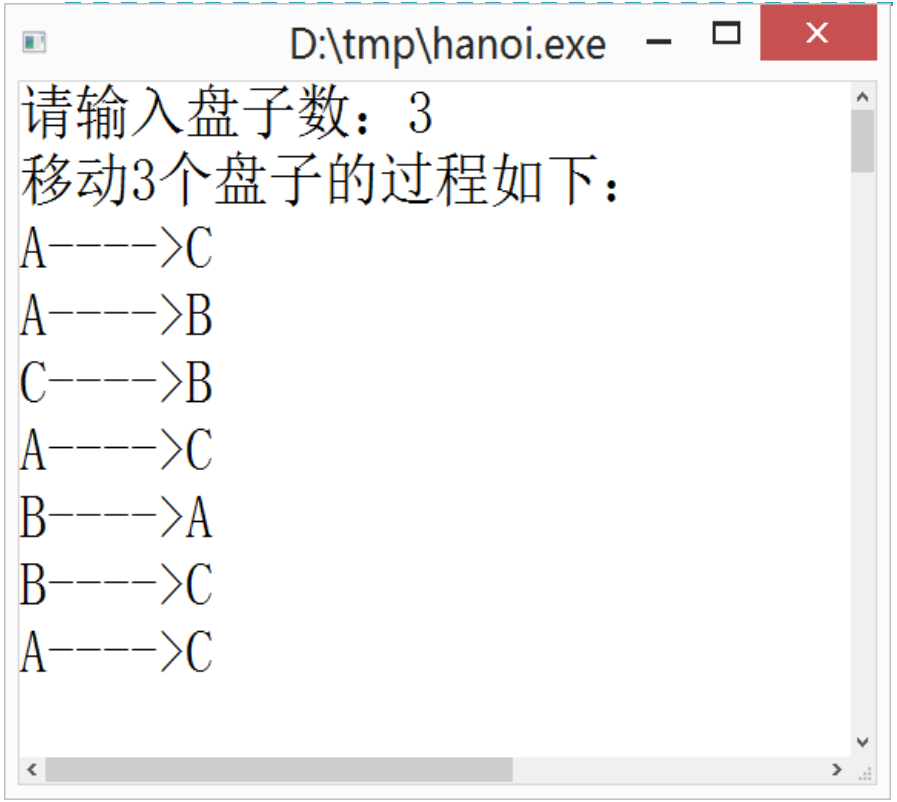
```
#include <iostream>
using namespace std;
//函数move将一个盘子从x移到y
void move(char x,char y)
{
  cout<<x<<"---->"<<y<<endl;
}
```

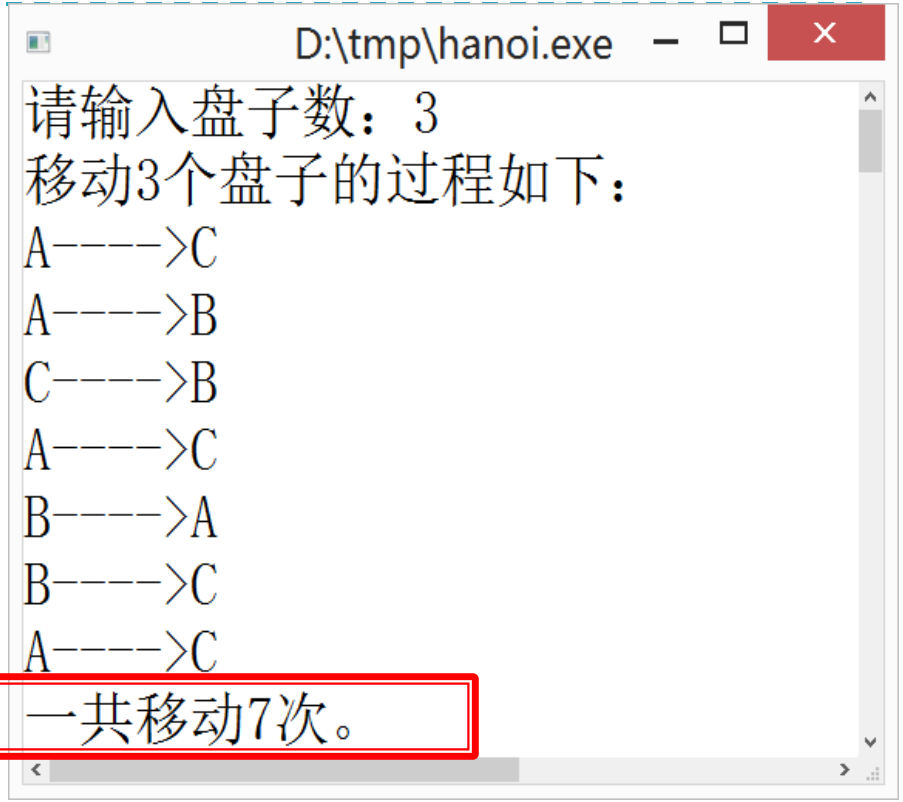
汉诺塔问题程序代码-hanoi函数

```
void hanoi(int n,char a,char b, char c)
{   if(n==1)
        move(a,c);
    else
    {  hanoi(n-1,a,c,b); //借助c将n-1个盘子从a移到b
       move(a,c); //从a移到c
       hanoi(n-1,b,a,c); //借助a将n-1个盘子从b移到c
    }
}
```

汉诺塔问题程序代码-main函数

```
int main()
{
    int m;
    cout<<"请输入盘子数：";
    cin>>m;
    cout<<"移动"<<m<<"个盘子的过程如下："<<endl;
    hanoi(m,'A','B','C');
    return 0;
}
```


D:\tmp\hanoi.exe
请输入盘子数：3
移动3个盘子的过程如下：
A----->C
A----->B
C----->B
A----->C
B----->A
B----->C
A----->C


D:\tmp\hanoi.exe
请输入盘子数：3
移动3个盘子的过程如下：
A----->C
A----->B
C----->B
A----->C
B----->A
B----->C
A----->C
一共移动7次。

如果需要输出总的移动次数，这时只需定义一个全局变量s，并在move函数中计数，即执行s++;即可。

```
int s=0;  //全局变量定义
void move(char x,char y)
{
    cout<<x<<"---->"<<y<<endl;
    s++;  //s统计移动的次数
}
```

主函数中添加：**cout<<"一共移动"<<s<<"次。"<<endl;**