

作业

```
// 汉诺塔问题是指：一块板上有三根针 A、B、C。
// A 针上套有 64 个大小不等的圆盘，按照大的在下、小的在上的顺序排列，
// 要把这 64 个圆盘从 A 针移动到 C 针上，每次只能移动一个圆盘，
// 移动过程可以借助 B 针。但在任何时候，任何针上的圆盘都必须保持大盘在下，小盘在上。
// 从键盘输入需移动的圆盘个数，给出移动的过程。
```

代码

算法思想

```
// 总体思想是
// 将 A 柱上的(n-1)个盘移动到 B 柱
// 然后将 A 柱上的最后一个盘移动到 C 柱
// 然后将 B 柱的(n-2)个盘移动到 C 柱
// 总结：递归实现
```

代码实现

```
// 程序的具体实现
void moveTo(char p_f, char p_s){
    printf("柱子%c最上层盘移动到柱子%c\n", p_f, p_s);
}
void P_Runing(char A, char B, char C, int n){
    if(n==1){//如果只有一个盘需要移动直接将 A 柱移动到 C 柱
        moveTo(A, C);
    }else{
        P_Runing(A, C, B, n-1); //将 A 柱(n-1)个盘移动到 B 柱
        moveTo(A, C); //将剩余 A 柱的一个盘移动到C
        P_Runing(B, A, C, n-1); //将 B 柱的(n-1)个盘移动到 C 柱
    }
}
int main(){
    char A = 'A', B = 'B', C = 'C';
    int n;
    scanf("%d", &n);
    P_Runing(A, B, C, n);
}
//console
4
柱子A最上层盘移动到柱子B
柱子A最上层盘移动到柱子C
柱子B最上层盘移动到柱子C
柱子A最上层盘移动到柱子B
柱子C最上层盘移动到柱子A
柱子C最上层盘移动到柱子B
柱子A最上层盘移动到柱子B
柱子A最上层盘移动到柱子C
柱子B最上层盘移动到柱子C
柱子B最上层盘移动到柱子A
柱子C最上层盘移动到柱子A
柱子B最上层盘移动到柱子C
```

柱子A最上层盘移动到柱子B
柱子A最上层盘移动到柱子C
柱子B最上层盘移动到柱子C

解析

```
P_Runing(A,C,B,n-1); //将 A 柱(n-1)个盘移动到 B 柱  
moveTo(A,C); //将剩余 A 柱的一个盘移动到C  
P_Runing(B,A,C,n-1); //将 B 柱的(n-1)个盘移动到 C 柱
```

这三行代码是实现汉诺塔的具体代码

代码中具体的实现步骤, `P_Runing(x1,x2,x3,N-1)` 函数的第一个参数为待移动柱, 第二个参数为辅助柱, 第三个参数为待接收柱. 因为在逐层的递归中待移动柱、辅助柱、接收柱是相对的

vn流程图



