**Tower of Hanoi**

**Jipa Bogdan Rares 421G**

The Tower of Hanoi is a mathematical game or puzzle consisting of three rods and a number of disks of various diameters, which can slide onto any rod. The puzzle begins with the disks stacked on one rod in order of decreasing size, the smallest at the top, thus approximating a conical shape. The objective of the puzzle is to move the entire stack to one of the other rods, obeying the following rules:

* Only one disk may be moved at a time.
* Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
* No disk may be placed on top of a disk that is smaller than it.

The minimal number of moves required to solve a Tower of Hanoi puzzle is **2n − 1**, where **n** is the number of disks.

In order to distinguish all three rods from each other, we will have to give them specific names:

* The first rod is **the source rod**
* The second rod is **the auxiliary rod**
* The third rod is **the destination rod**

A close-up of a toy

Description automatically generated

Image via [Wikipedia](https://en.wikipedia.org/wiki/Tower_of_Hanoi)

As these names suggests , we have to move the disks from the source rod to the destination rod.

For example, when n = 3, solving will require 7 steps to transfer all three disks from the source. These 7 steps are as follows:

Move disk 1 from the source to the destination.

Move disk 2 from the source to the auxiliary.

Move disk 1 from the destination to the auxiliary.

Move disk 3 from the source to the destination.

Move disk 1 from the auxiliary to the source.

Move disk 2 from the auxiliary to the destination.

Move disk 1 from the source to the destination.

One important thing to mention is that we have make two cases:

* one for n being an even number
* the other for n being an odd number.

For n being an even number, there are 3 transfers being repeated in an order:

* First transfer between source and auxiliary pole
* Second transfer between source and destination pole
* Third transfer between auxiliary and destination pole

And, for n being an even number the 3 transfers being repeated in an order are:

* First transfer between source and destination pole
* Second transfer between source and auxiliary pole
* Third transfer between destination and auxiliary pole

Code:

#include <iostream>  
#include <stack>  
#include <cmath>  
using namespace std;  
//  
int transfer\_disk(stack<int>& a,stack<int>& b){  
 if(b.empty()==true){  
 b.push(a.top());  
 a.pop();  
 return 1; }  
 else if(a.empty()==true){  
 a.push(b.top());  
 b.pop();  
 return 2;  
 }  
 else{ if(b.top()>a.top()){  
 b.push(a.top());  
 a.pop();  
 return 1;  
 }  
 else{  
 a.push(b.top());  
 b.pop();  
 return 2;  
 }  
 }  
}  
void push\_number\_of\_disk(stack<int> &a, int nrdisks){  
 for(int i=nrdisks;i>=1;i--){  
 a.push(i);  
 }  
}  
int main(){  
 stack<int> source,aux,destination;  
 int n=0;  
 cout<<"How many disks do you want to insert?"<<endl;cin>>n;cout<<endl;  
 push\_number\_of\_disk(source,n);  
 int total\_moves=pow(2,n)-1;  
 int i=1;  
  
if(n%2==0){  
 while(i<=total\_moves){  
 if(i%3==1){  
 int y=transfer\_disk(source,aux);  
 if(y==1){  
 cout<<"Move the disk "<<aux.top()<<" from source to auxiliary"<<endl;  
 }  
 else  
 cout<<"Move the disk "<<source.top()<<" from auxiliary to source"<<endl;  
 }  
 else if(i%3==2){  
 int y=transfer\_disk(source,destination);  
 if(y==1){  
 cout<<"Move the disk "<<destination.top()<<" from source to destination"<<endl;  
 }  
 else  
 cout<<"Move the disk "<<source.top()<<" from destination to source"<<endl;  
 }  
 else{  
 int y=transfer\_disk(aux,destination);  
 if(y==1){  
 cout<<"Move the disk "<<destination.top()<<" from auxiliary to destination"<<endl;  
 }  
 else  
 cout<<"Move the disk "<<aux.top()<<" from destination to auxiliary"<<endl;  
 }  
 i++;  
 }  
}  
else{  
 while(i<=total\_moves){  
 if(i%3==1){  
 int y=transfer\_disk(source,destination);  
 if(y==1){  
 cout<<"Move the disk "<<destination.top()<<" from source to destination"<<endl;  
 }  
 else  
 cout<<"Move the disk "<<source.top()<<" from destination to source"<<endl;  
 }  
 else if(i%3==2){  
 int y=transfer\_disk(source,aux);  
 if(y==1){  
 cout<<"Move the disk "<<aux.top()<<" from source to auxiliary"<<endl;  
 }  
 else  
 cout<<"Move the disk "<<source.top()<<" from auxiliary to source"<<endl;  
 }  
 else{  
 int y=transfer\_disk(aux,destination);  
 if(y==1){  
 cout<<"Move the disk "<<destination.top()<<" from auxiliary to destination"<<endl;  
 }  
 else  
 cout<<"Move the disk "<<aux.top()<<" from destination to auxiliary"<<endl;  
 }  
 i++;  
 }  
}

cout<<endl<<"Final outcome:"<<endl;  
while(destination.empty()!=true){  
 cout<<destination.top()<<endl;  
 destination.pop();  
 }  
  
  
return 0;  
}

Screenshots of examples:

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated