Recursion

1. Tower of Hanoi
Uses a recursive function to solve the Tower of Hanoi problem – only five lines of code!

```cpp
/*
 * This is the main.cpp file
 * Copyright 2010 Enoch Hwang
 */
#include <iostream>
#include "Hanoi.h"
using namespace std;

int main (int argc, char * const argv[]) {
  int ndisks;
  int frompeg = 1;
  int temppeg = 2;
  int topeg = 3;

  cout << "Enter the number of disks you want? ";
  cin >> ndisks;
  move(ndisks, frompeg, temppeg, topeg);
  return 0;
}

void move(int ndisks, int frompeg, int temppeg, int topeg);
```

```cpp
/*
 * Hanoi.h
 * Copyright 2010 Enoch Hwang
 */
void move(int ndisks, int frompeg, int temppeg, int topeg);

/*
 * Hanoi.cpp
 * Copyright 2010 Enoch Hwang
 */
#include <iostream>
#include "Hanoi.h"
using namespace std;

void move(int ndisks, int frompeg, int temppeg, int topeg){
  if(ndisks > 1){
    move(ndisks-1, frompeg, topeg, temppeg);
    cout << "A-move disk " << ndisks << " from peg " << frompeg << " to peg " << topeg << endl;
    move(ndisks-1, temppeg, frompeg, topeg);
  }else{
    cout << "B-move disk " << ndisks << " from peg " << frompeg << " to peg " << topeg << endl;
  }
}
```
Sample output

Enter the number of disks you want? 3
B-move disk 1 from peg 1 to peg 3
A-move disk 2 from peg 1 to peg 2
B-move disk 1 from peg 3 to peg 2
A-move disk 3 from peg 1 to peg 3
B-move disk 1 from peg 2 to peg 1
A-move disk 2 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3

Enter the number of disks you want? 5
B-move disk 1 from peg 1 to peg 3
A-move disk 2 from peg 1 to peg 2
B-move disk 1 from peg 3 to peg 2
A-move disk 3 from peg 1 to peg 3
B-move disk 1 from peg 2 to peg 1
A-move disk 2 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3
A-move disk 4 from peg 1 to peg 2
B-move disk 1 from peg 3 to peg 2
A-move disk 2 from peg 3 to peg 1
B-move disk 1 from peg 2 to peg 1
A-move disk 3 from peg 2 to peg 1
B-move disk 1 from peg 3 to peg 2
A-move disk 2 from peg 1 to peg 2
B-move disk 1 from peg 3 to peg 2
A-move disk 3 from peg 1 to peg 3
B-move disk 1 from peg 2 to peg 1
A-move disk 2 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3
A-move disk 4 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3
A-move disk 2 from peg 1 to peg 2
B-move disk 1 from peg 3 to peg 2
A-move disk 3 from peg 1 to peg 3
B-move disk 1 from peg 2 to peg 1
A-move disk 2 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3
A-move disk 4 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3
A-move disk 2 from peg 1 to peg 2
B-move disk 1 from peg 3 to peg 2
A-move disk 3 from peg 1 to peg 3
B-move disk 1 from peg 2 to peg 1
A-move disk 2 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3
A-move disk 4 from peg 2 to peg 3
B-move disk 1 from peg 1 to peg 3

2. Exercise

Modify the program so that for each move step, it will also print out the correct move number.