THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

Artificial Intelligence (CS13217)

Lab Report 2

Name: Kanza Afzal	
Registration #:	CSU-XS18-132
Lab Report #:	02
Dated:	16-04-2018
Submitted To:	Mr. Usman Ahmed

Ťhe University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

Experiment # 2
Implementing Tower of Hanoi Problem

Objective
To understand and implement the Tower of Hanoi Problem.

Software Tool
1.
Dev

1 Theory

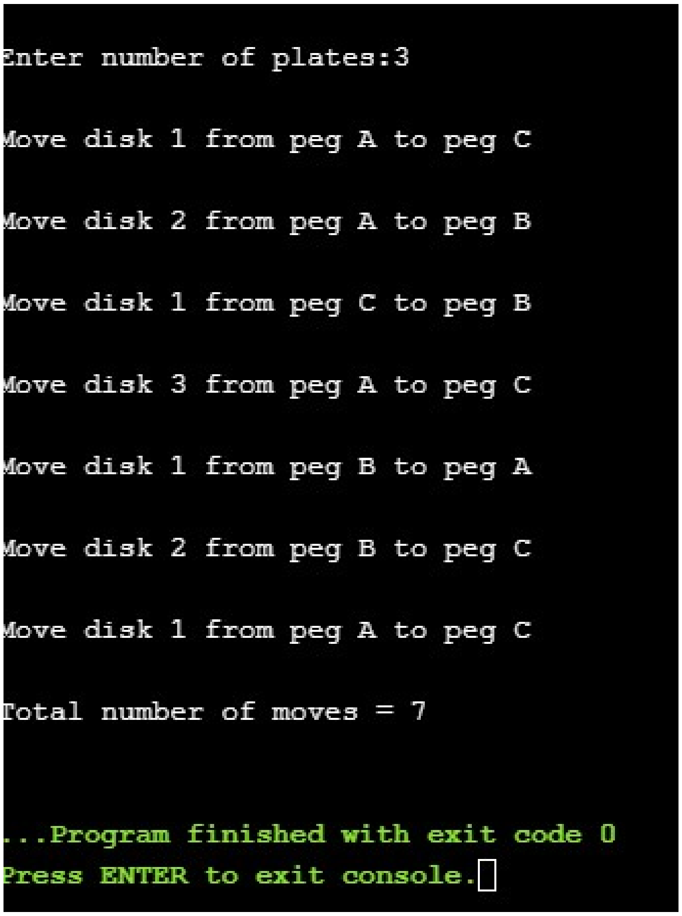
The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.
The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:
1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk.

With three disks, the puzzle can be solved in seven moves. The minimum number of moves required to solve a Tower of Hanoi puzzle is 2n - 1, where n is the number of disks.

2 Task

2.1 Procedure: Task 1

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



```
Enter number of plates:3

Move disk 1 from peg A to peg C

Move disk 2 from peg A to peg B

Move disk 1 from peg C to peg B

Move disk 3 from peg A to peg C

Move disk 1 from peg B to peg A

Move disk 2 from peg B to peg C

Move disk 1 from peg A to peg C

Total number of moves = 7


...Program finished with exit code 0
Press ENTER to exit console.
```

Figure 1: Tower of Hanoi output

2.2 Procedure: Task 2

```
int TOH(int,char,char,char);
 int main()
 {
   int n;
   printf(”\nEnter␣number␣of␣plates:”);
   scanf(”%d”,&n);
    int c = TOH(n,’A’,’C’,’B’);
    printf(”\n”);
    printf(”Total␣number␣of␣moves␣=␣%d␣\n␣”, c);
  return 0;
  }
int TOH(int n,char x,char y,char z)
{
   int count = 0;
   if(n>0){
       count = TOH(n−1, x, z, y);
       printf(”\nMove␣disk␣%d␣from␣peg␣%c␣to␣peg␣%c\n”, n, x, y);
       count++;
       count += TOH(n−1, z, y, x) ;
   }
   return count;
}
```

3 Conclusion

When the number of disks is 2 the number of moves it takes is 3, for 4 disks 15 and for 64 disks the program keeps running infinity loop