VNR VIGNANA JYOTI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Affiliated to J.N.T.U, Hyderabad) Bachupally(v), Hyderabad, Telangana, India

CSBS

SEMESTER - II

COURSE BASED PROJECT REPORT

DATA STRUCTURES AND ALGORITHMS

Under the Guidance of:

Mrs. S Nagini

HOD, Dept. of CSE & CSBS

CERTIFICATE

This is to certify that

21071A3244-M.V.Vasavi Swathi
21071A3245-Mirthipati Sravanth
21071A3246-Mogili Rishitha Sagar
21071A3247-Mohammed Abdul Sami
21071A3248-Sirish

have completed their course project work at CSE Department of VNR VJIET, Hyderabad entitled "_____" in complete fulfilment of the requirements for the award of B.Tech degree during the academic year 2021-2022.

The performance of the Team was

_____

. and He/she worked well as a part of a Team.

Dr. S. Nagini
Head of the Department, CSE & CSBS
VNRVJIET

The **Tower of Hanoi** is a mathematical game or puzzle consisting of three rods and several 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 the last rod, obeying the following rules:

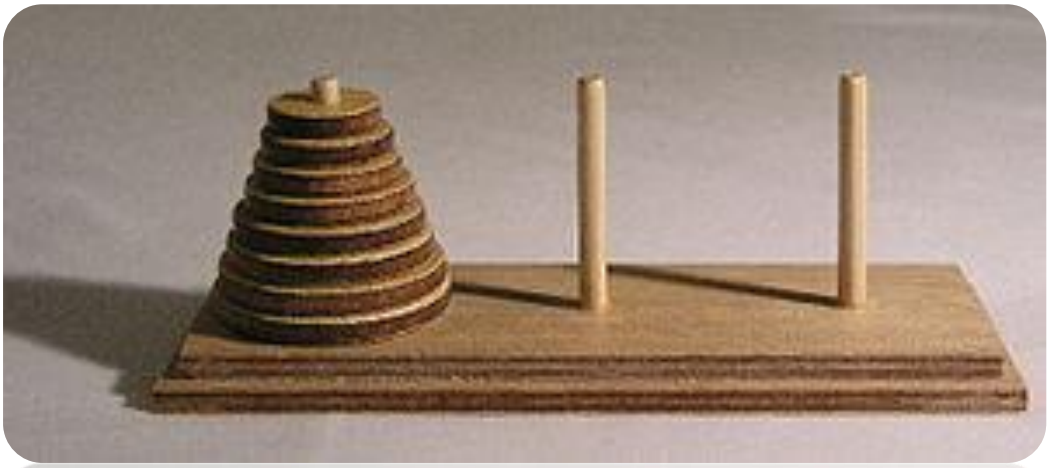
1.Only one disk may be moved at a time.

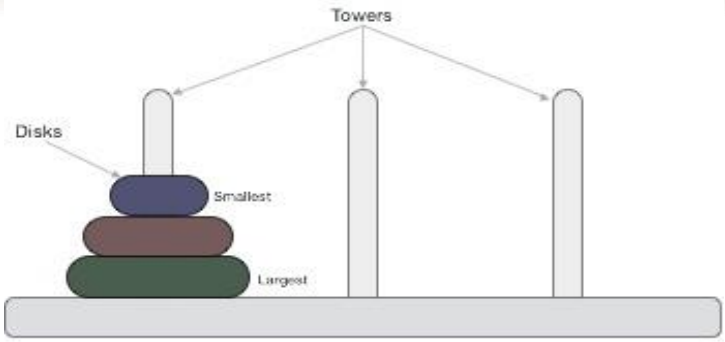
2.Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.

3.No disk may be placed on top of a disk that is smaller than it.

★ RULES :

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





ALGORITHM (TOWER OF HANOII)

```
START
 Procedure TOH(disk, source, dest, aux)
 IF disk == 1, THEN
      move disk from source to dest
   ELSE
      TOH(disk - 1, source, aux, dest)     // Step 1
      moveDisk(source to dest)          // Step 2
      TOH(disk - 1, aux, dest, source)     // Step 3
   END IF
 END Procedure
 STOP
```

CODE FOR TOWERS OF HANOII SIMULATION

```
#include<graphics.h>
#include<dos.h>
#include<math.h>
#include<conio.h>
#include<stdio.h>

int tower[3][100];
int top[3];
int from,to;
int diskinair;
int l,b,u;
int pushcnt=0;
int  popcnt=0;


void push(int to,int diskno)
//putting disk on tower
{
tower[to-1][++top[to-1]]=diskno;
pushcnt++;
}


int pop(int from)
//take topmost disk from tower
{
  popcnt=popcnt+1;
 return(tower[from-1][top[from-1]--]);
  }
```

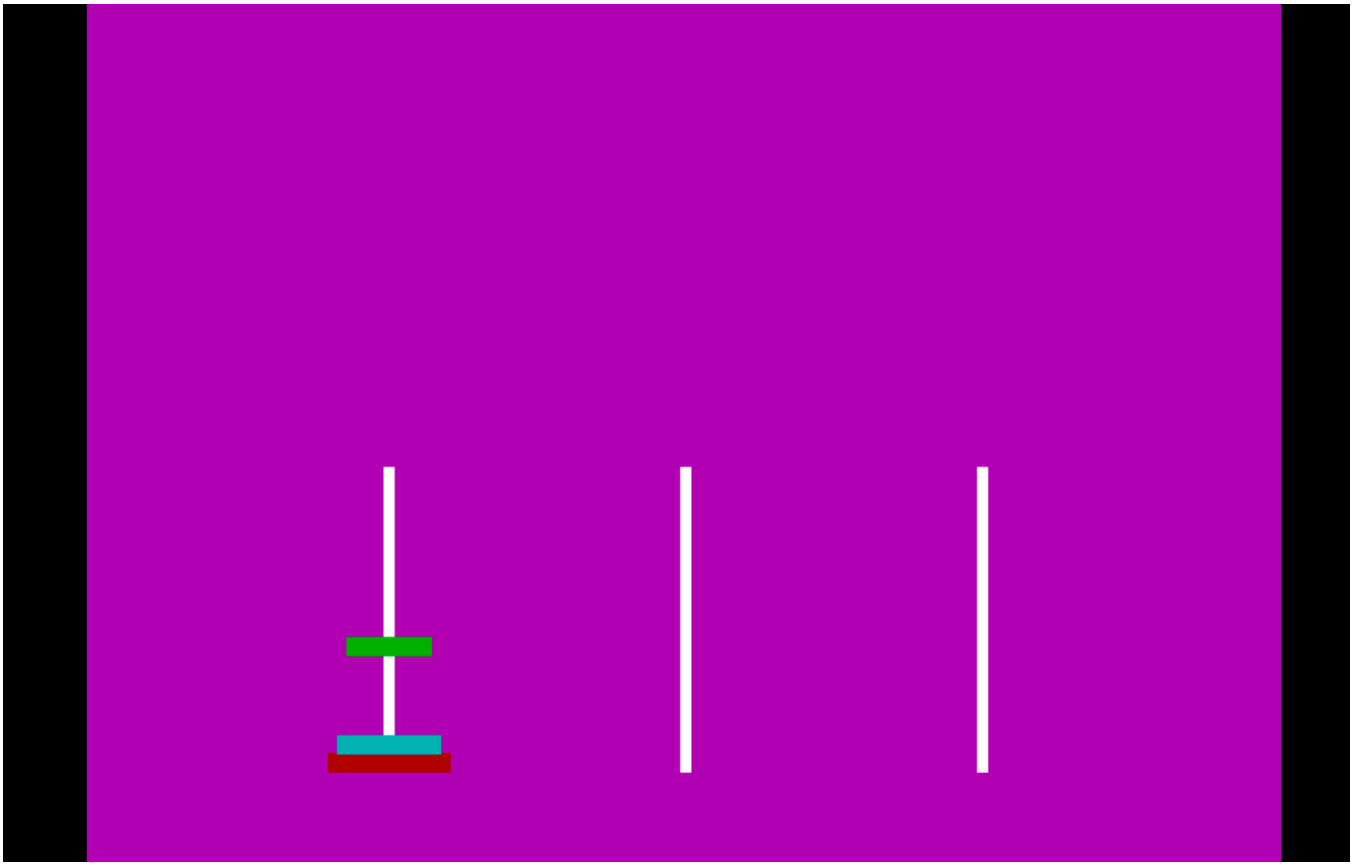
```
void drawStill()
 {
 int j,i,disk;
 cleardevice();
 for(j=1;j<=3;j++)
 {
   //draw tower
   setfillstyle(1,WHITE);
     bar(j*l,u,j*l+5,b);
   //draw all disks on tower
   for(i=0;i<=top[j-1];i++)
   {
     disk=tower[j-1][i];
     setfillstyle(SOLID_FILL,1+disk);
     bar(j*l-15-disk*5,b-(i+1)*10,j*l+5+15+disk*5,b-i*10);
     }
    }
  }

void animator()
{
 //to show the movement of disk
 int x,y,dif,sign;
 diskinair=pop(from);
 x=from*l;
 y=b-(top[from-1]+1)*10;
 //taking disk upward from the tower
 for(;y>u-20;y-=15)
 {
   drawStill();
   setfillstyle(SOLID_FILL,1+diskinair);
   bar(x-15-diskinair*5,y-10,x+5+15+diskinair*5,y);
   delay(100);
 }
```

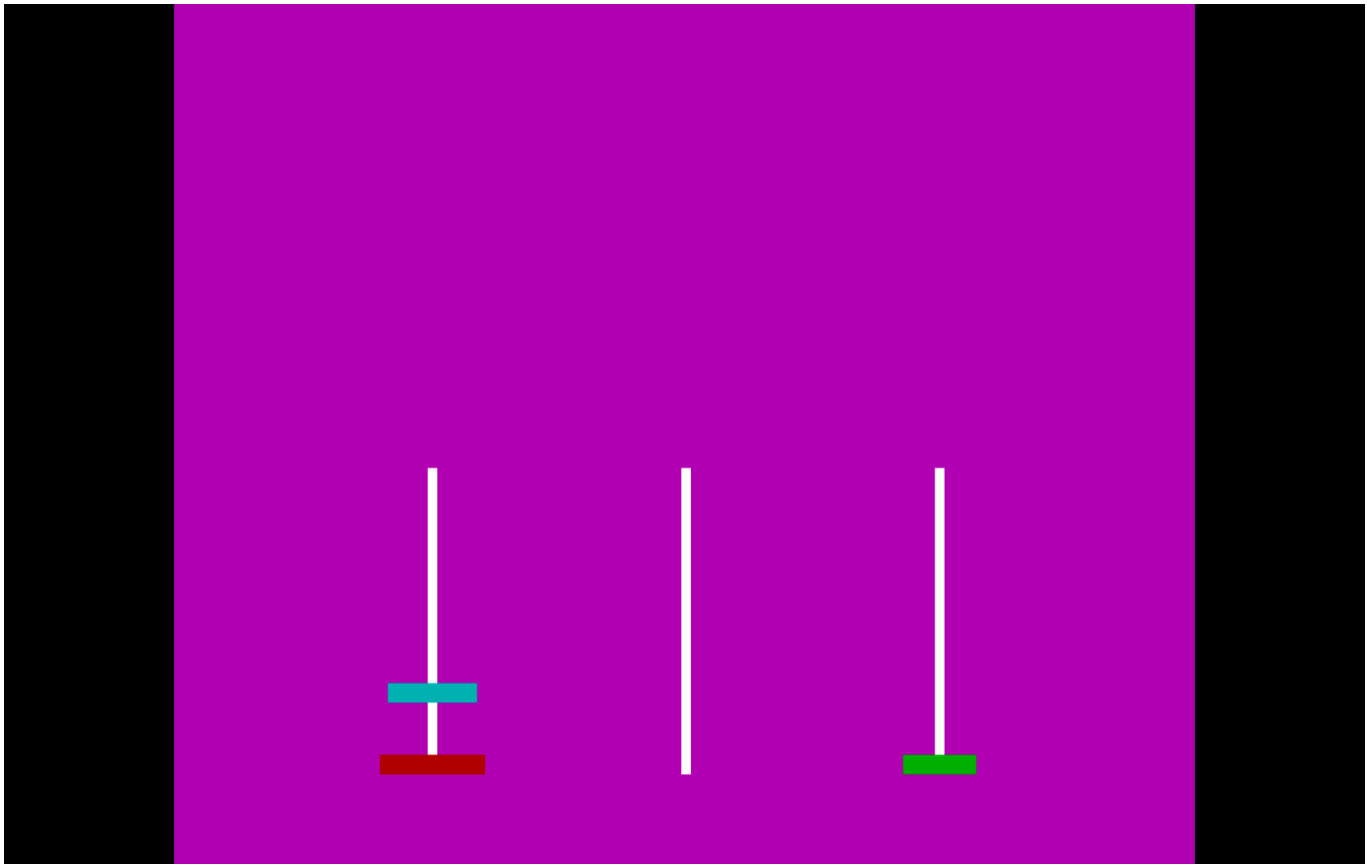
```
 y=u-20;
  dif=to*l-x;
  sign=dif/abs(dif);
  //moving disk toward a target tower
  for(;abs(x-to*l)>25;x+=sign*15)
  {
     drawStill();
     setfillstyle(SOLID_FILL,1+diskinair);
     bar(x-15-diskinair*5,y-10,x+5+15+diskinair*5,y);
     delay(100);
   }
   x=to*l;
  for(;y<b-(top[to-1]+1)*10;y+=15)
  {
   drawStill();
   setfillstyle(SOLID_FILL,1+diskinair);
   bar(x-15-diskinair*5,y-10,x+5+15+diskinair*5,y);
   delay(100);
   }
   push(to,diskinair);
   drawStill();
   }
void moveTopN(int n,int a,int b,int c)
{
  //move top n disk from tower 'a' to 'b'used for swapping
  if(n>=1)
  {
    moveTopN(n-1,a,c,b);
    drawStill() ;
    delay(100);
    from=a;
    to=c;
    //animating the move
    animator();
    moveTopN(n-1,b,a,c);
     }
   }
```

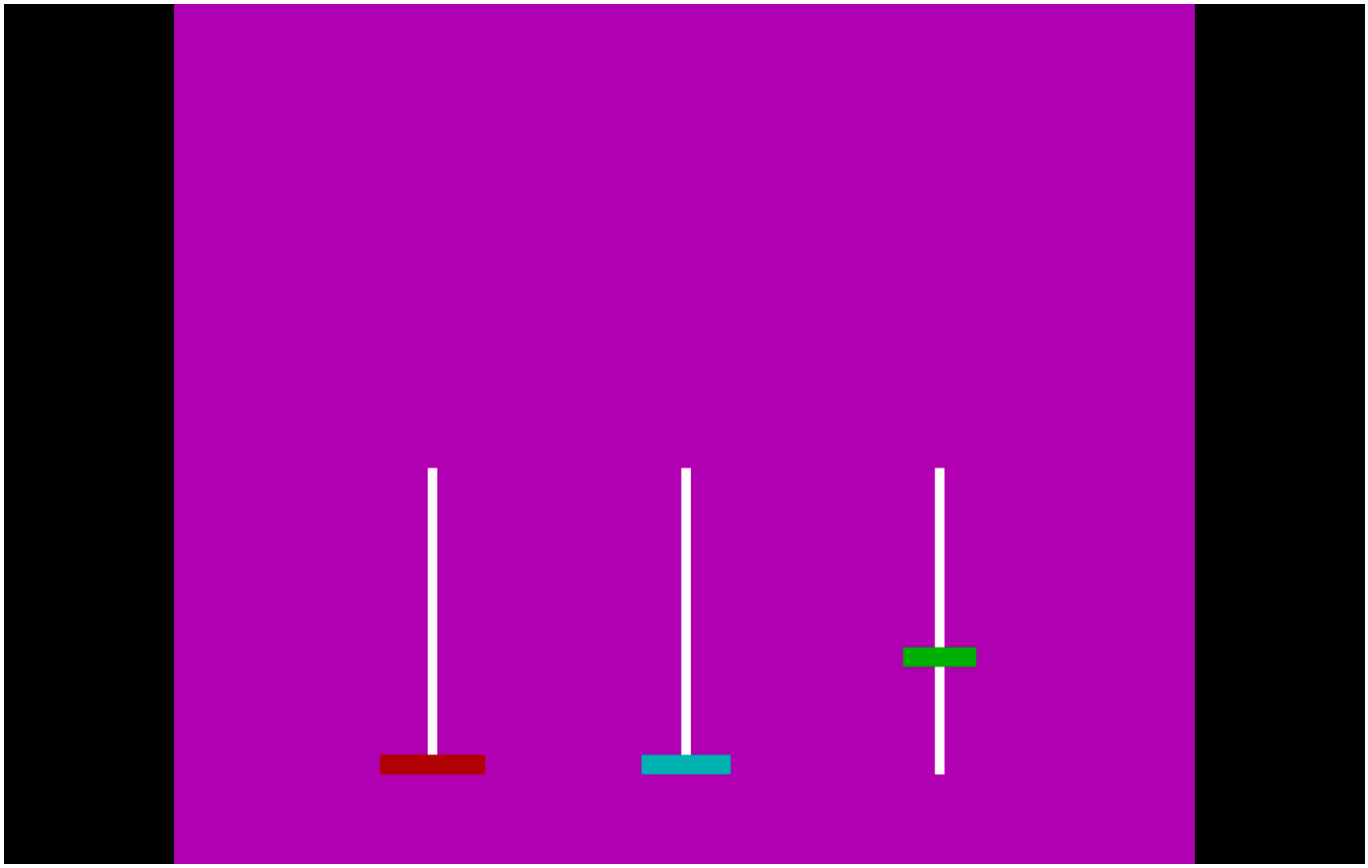
```
void main()
 {
  int po,i,gd=DETECT,gm,n;
  clrscr();
  printf("Enter number of disks\n");
  scanf("%d",&n);
  po=pow(2,n);
  po=po-1;
   initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
  setbkcolor(MAGENTA);
 // setConsoleTextAttribute(GetStdHandle(),WHITE);
  //setting all towers empty
  for(i=0;i<3;i++)
  {
    top[i]=-1;
   }
 //putting all disks on tower 'a'
 for(i=n;i>0;i--)
 {
   push(1,i);
   }
 l=getmaxx()/4;
 b=getmaxy()-50;
 u=getmaxy()/3+100;
 //start solving
 moveTopN(n,1,2,3);
 delay(100);
  printf("No. of times push operation is performed:\n%d \nNo.of times pop
operation is performed:\n%d",pushcnt,popcnt);
  printf("\nTotal no, of Moves Done:\n%d",po);
  getch();
}
```

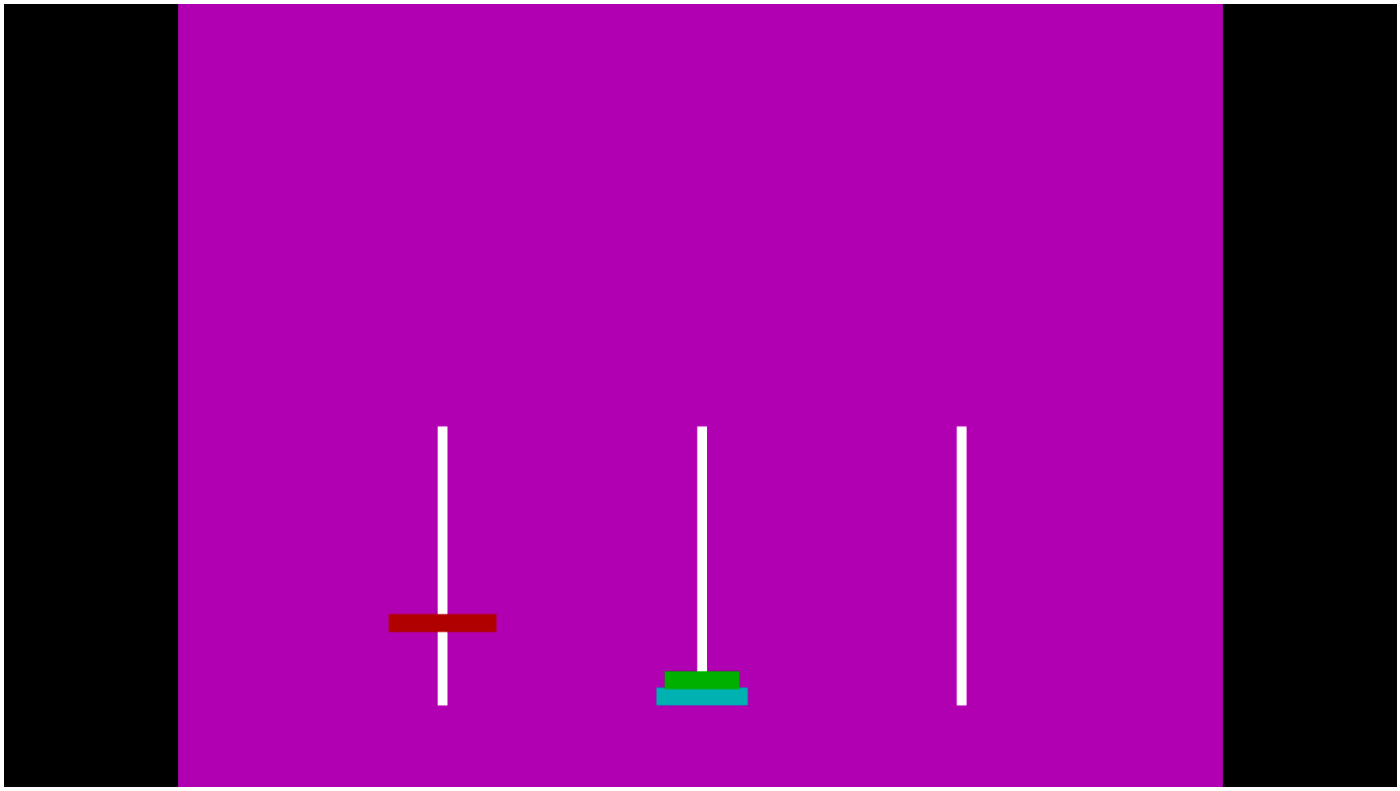
★ OUTPUT :

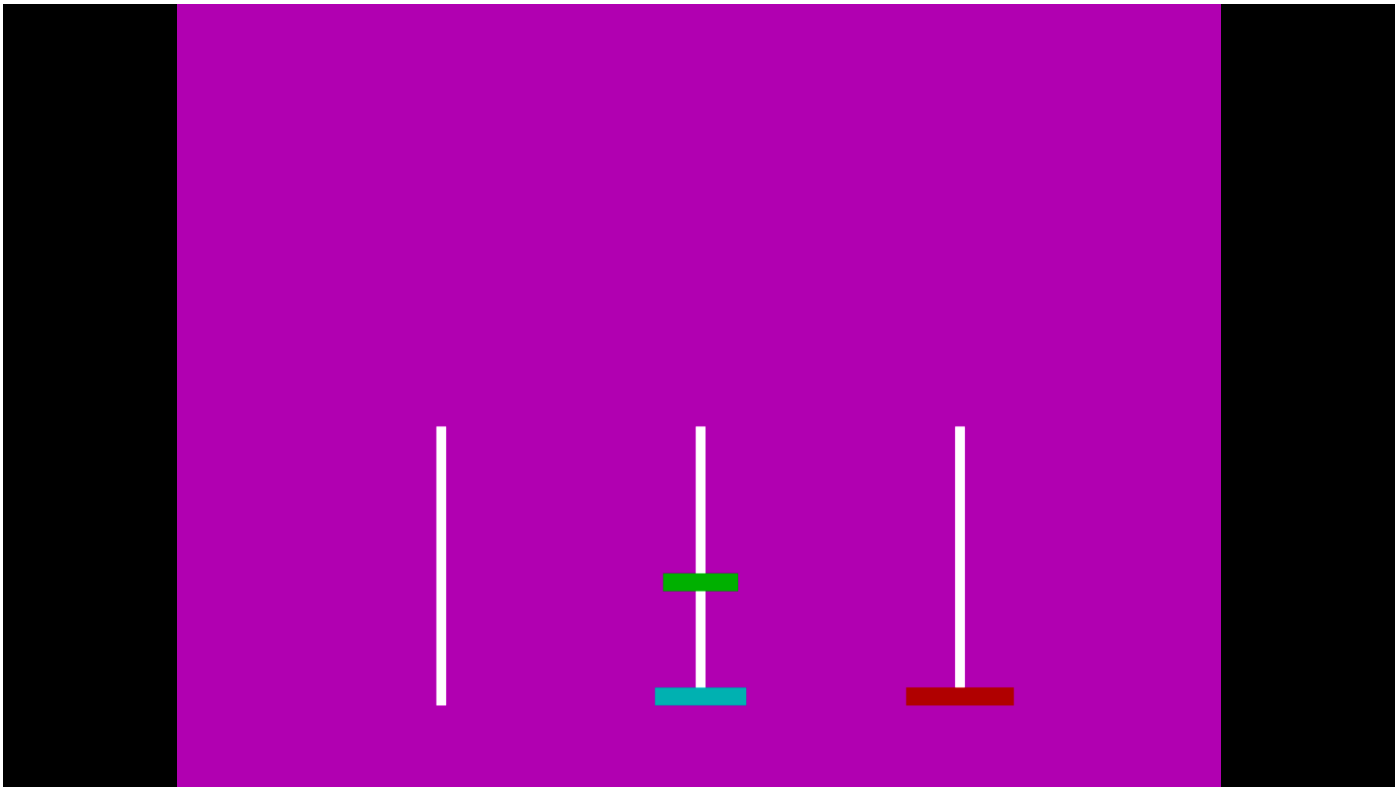
```
Enter number of disks
3_
```

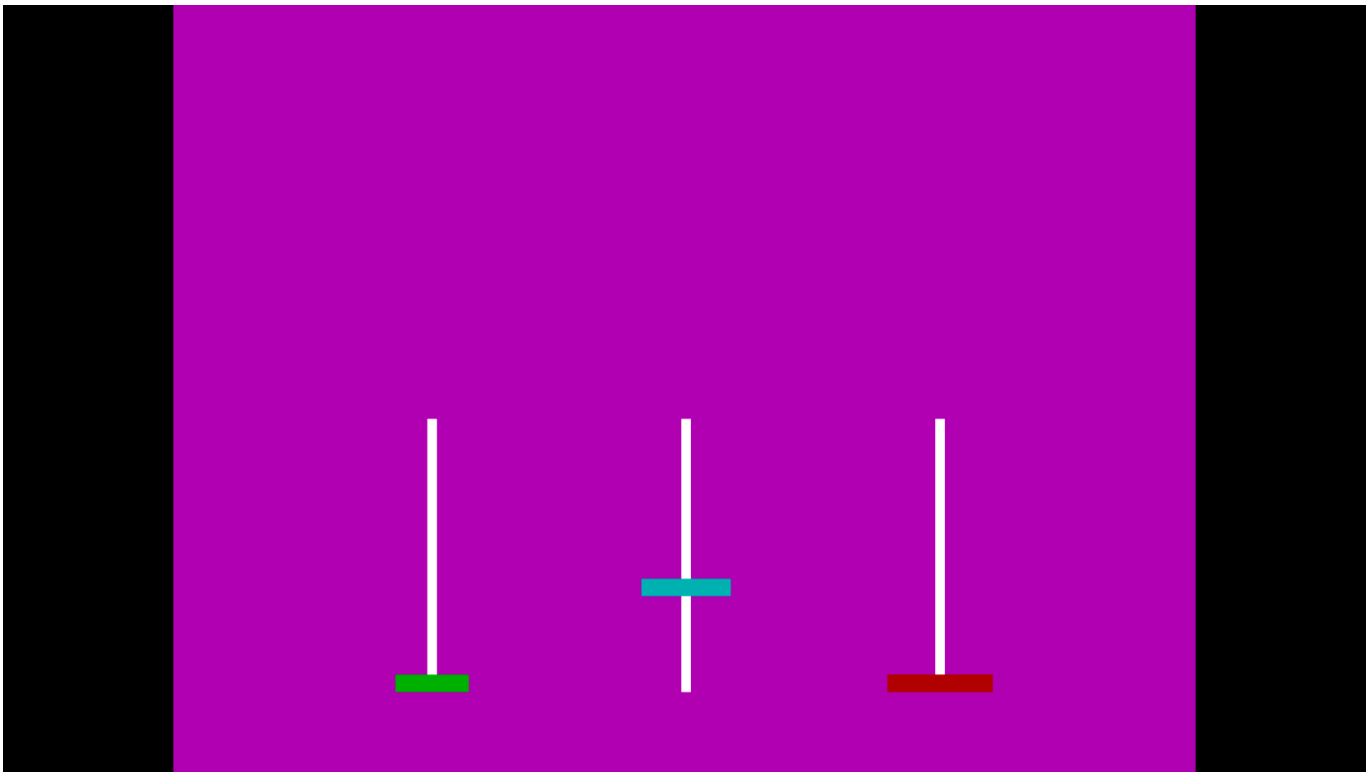


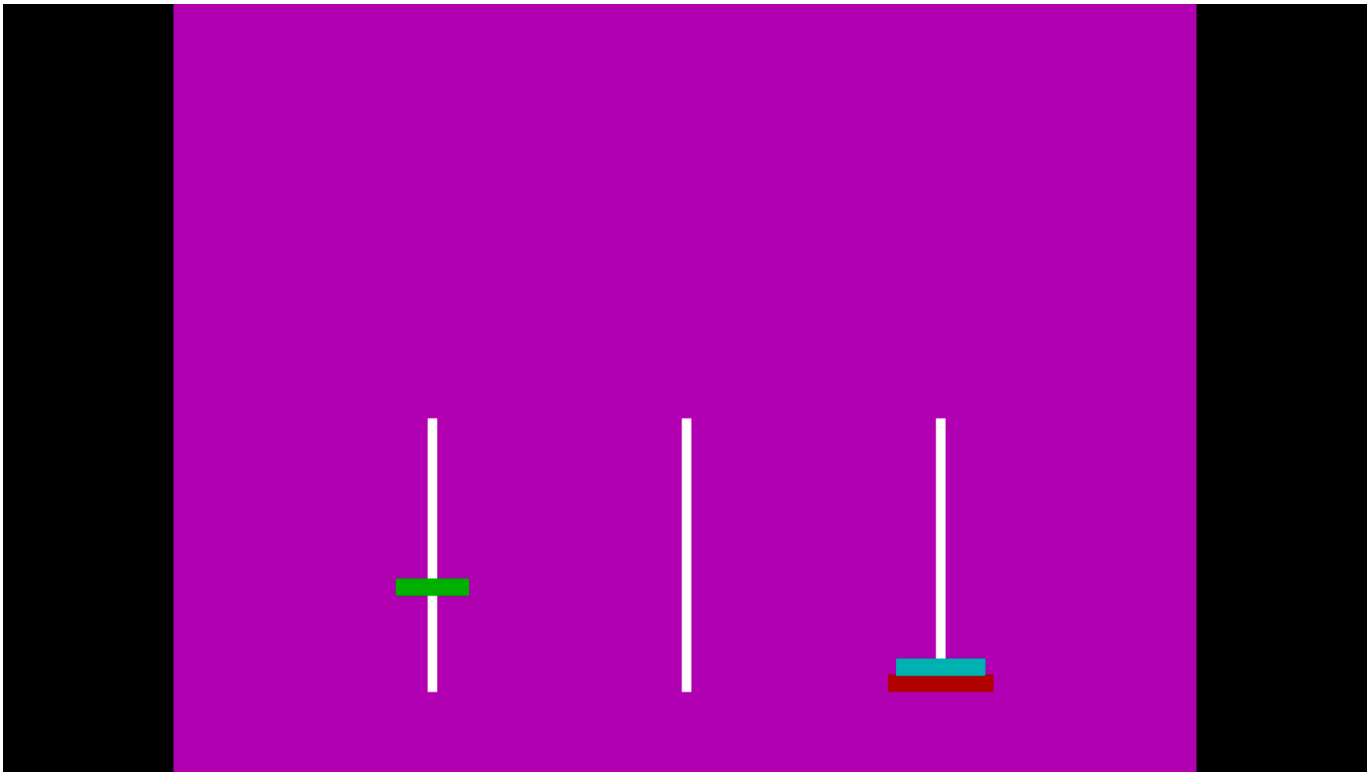


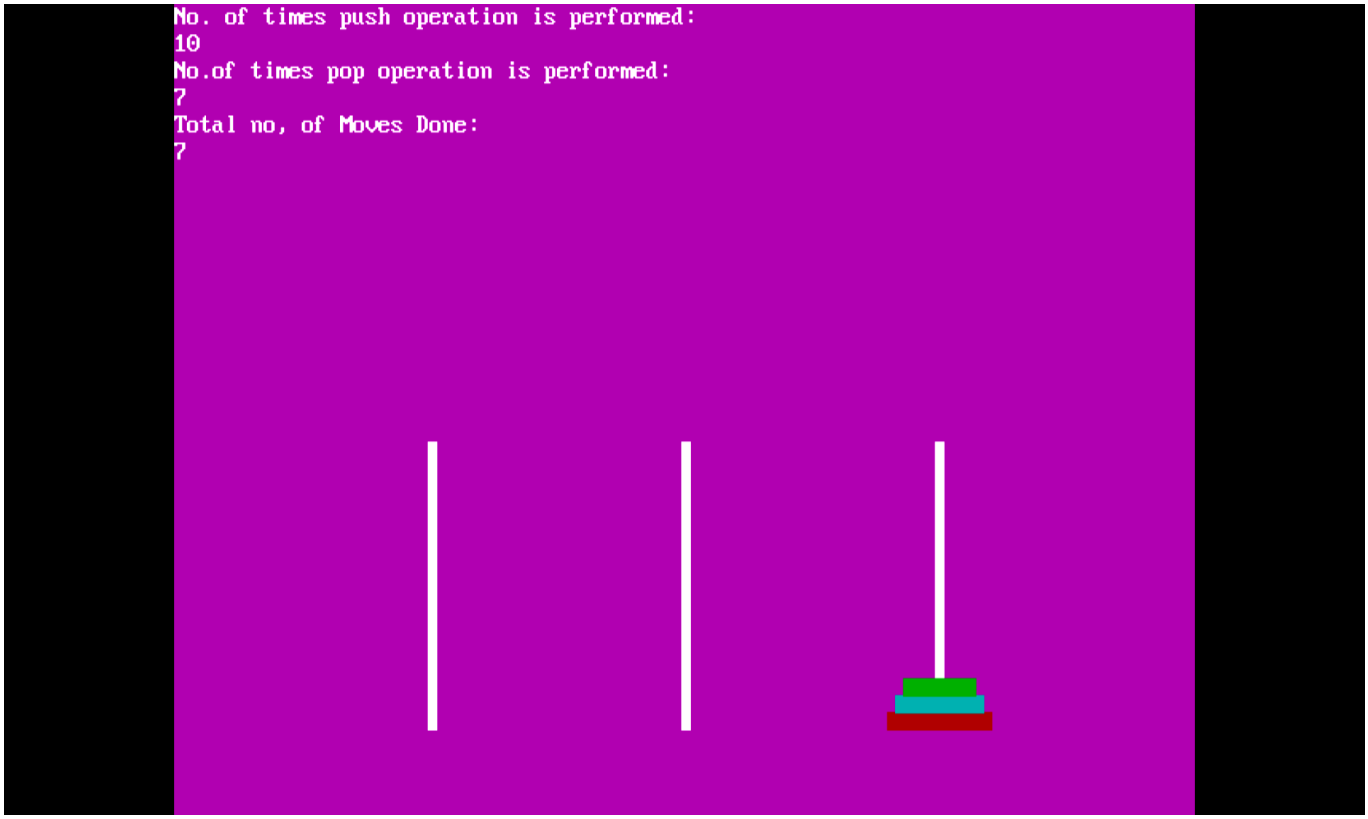











No. of times push operation is performed:
10
No.of times pop operation is performed:
7
Total no, of Moves Done:
7