**Data Structure**

**Towers of Hanoi**

CODE:

#include<stdio.h>

#include<iostream>

#include<math.h>

using namespace std;

struct node1

{

int data1;

node1 \*next1;

}\*top1 = NULL, \*p1 = NULL, \*np1 = NULL;

struct node2

{

int data2;

node2 \*next2;

}\*top2 = NULL, \*p2 = NULL, \*np2 = NULL;

struct node3

{

int data3;

node3 \*next3;

}\*top3 = NULL, \*p3 = NULL, \*np3 = NULL;

void push1(int data)

{

np1 = new node1;

np1->data1 = data;

np1->next1 = NULL;

if (top1 == NULL)

{

top1 = np1;

}

else

{

np1->next1 = top1;

top1 = np1;

}

}

int pop1()

{

int b = 999;

if (top1 == NULL)

{

return b;

}

else

{

p1 = top1;

top1 = top1->next1;

return(p1->data1);

delete(p1);

}

}

void push2(int data)

{

np2 = new node2;

np2->data2 = data;

np2->next2 = NULL;

if (top2 == NULL)

{

top2 = np2;

}

else

{

np2->next2 = top2;

top2 = np2;

}

}

int pop2()

{

int b = 999;

if (top2 == NULL)

{

return b;

}

else

{

p2 = top2;

top2 = top2->next2;

return(p2->data2);

delete(p2);

}

}

void push3(int data)

{

np3 = new node3;

np3->data3 = data;

np3->next3 = NULL;

if (top3 == NULL)

{

top3 = np3;

}

else

{

np3->next3 = top3;

top3 = np3;

}

}

int pop3()

{

int b = 999;

if (top3 == NULL)

{

return b;

}

else

{

p3 = top3;

top3 = top3->next3;

return(p3->data3);

delete(p3);

}

}

int top\_of\_stack()

{

if (top1 != NULL && top1->data1 == 1 )

{

return 1;

}

else if (top2 != NULL && top2->data2 == 1)

{

return 2;

}

else if (top3 != NULL && top3->data3 == 1)

{

return 3;

}

}

void display1()

{

cout<<endl;

node1 \*p1;

p1 = top1;

cout<<"Tower1-> "<<"t ";

while (p1 != NULL)

{

cout<<p1->data1<<"t ";

p1 = p1->next1;

}

cout<<endl;

}

void display2()

{

node2 \*p2;

p2 = top2;

cout<<"Tower2-> "<<"t ";

while (p2 != NULL)

{

cout<<p2->data2<<"t ";

p2 = p2->next2;

}

cout<<endl;

}

void display3()

{

node3 \*p3;

p3 = top3;

cout<<"Tower3-> "<<"t ";

while (p3 != NULL)

{

cout<<p3->data3<<"t ";

p3 = p3->next3;

}

cout<<endl;

cout<<endl;

}

void toh(int n)

{

int i, x, a, b;

for (i = 0; i < (pow(2,n)); i++)

{

display1();

display2();

display3();

x = top\_of\_stack();

if (i % 2 == 0)

{

if (x == 1)

{

push3(pop1());

}

else if (x == 2)

{

push1(pop2());

}

else if (x == 3)

{

push2(pop3());

}

}

else

{

if (x == 1)

{

a = pop2();

b = pop3();

if (a < b && b != 999)

{

push3(b);

push3(a);

}

else if (a > b && a != 999)

{

push2(a);

push2(b);

}

else if (b == 999)

{

push3(a);

}

else if (a == 999)

{

push2(b);

}

}

else if (x == 2)

{

a = pop1();

b = pop3();

if (a < b && b != 999)

{

push3(b);

push3(a);

}

else if (a > b && a != 999)

{

push1(a);

push1(b);

}

else if (b == 999)

{

push3(a);

}

else if (a == 999)

{

push1(b);

}

}

else if (x == 3)

{

a = pop1();

b = pop2();

if (a < b && b != 999)

{

push2(b);

push2(a);

}

else if (a > b && a != 999)

{

push1(a);

push1(b);

}

else if (b == 999)

{

push2(a);

}

else if (a == 999)

{

push1(b);

}

}

}

}

}

int main()

{

int n, i;

cout<<"enter the number of disks\n";

cin>>n;

for (i = n; i >= 1; i--)

{

push1(i);

}

toh(n);

return 0;

}

OUTPUT:

