**Towers of Hanoi problem:**

* It is a mathematical puzzle.
* It involves Moving disks from one pole to another pole.
* It takes the help of temporary(auxiliary) pole



**Rules:**

* Only one disk can be moved at a time.
* Each move consists of taking upper disk from the pole(STACK – LIFO) and placing on top of another Pole.
* No disk may be placed on top of smaller disk
* Process of moving as shown in diagram

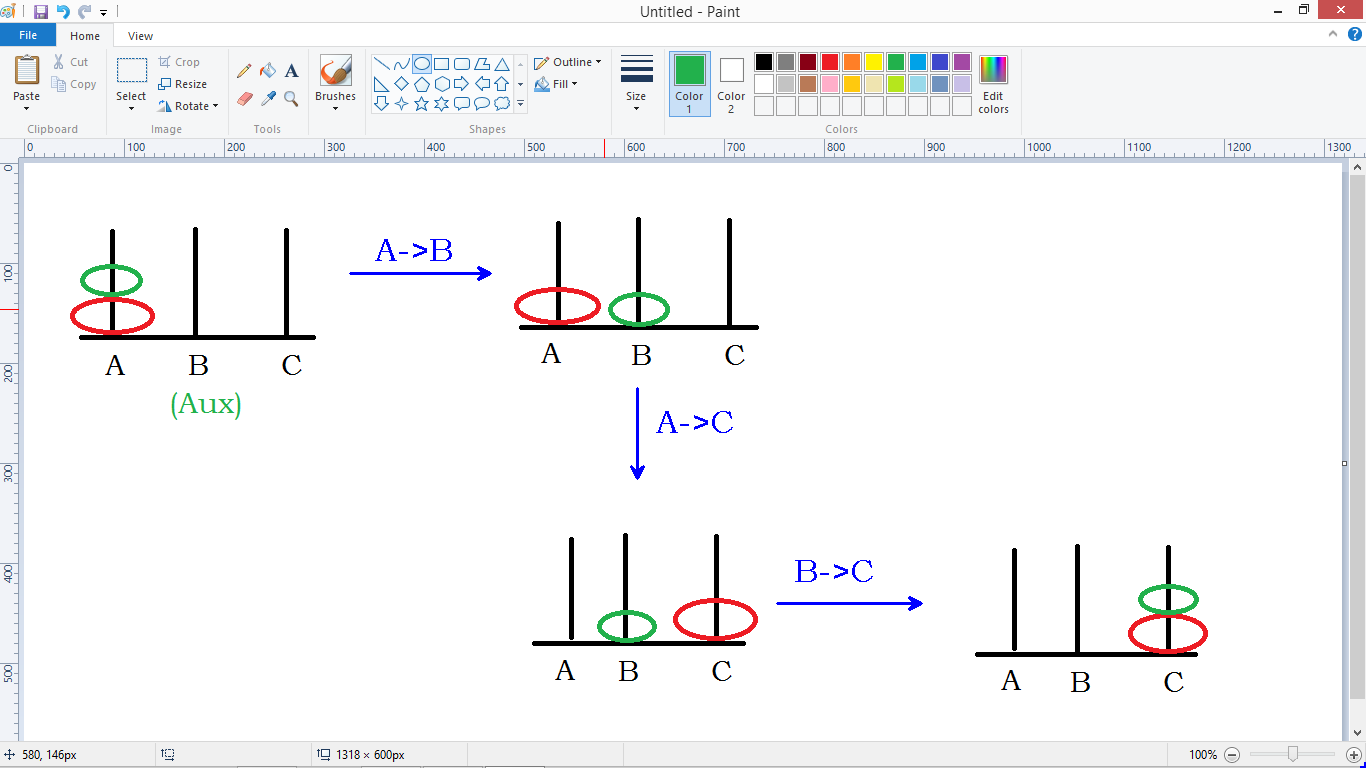
**Note:** If we have ‘n’ disks, ‘2^n-1’ moves required.

N=2 🡪 2^2 -1 🡪 3 moves required

N=3 🡪 2^3-1 🡪 7 moves required

…..

2 Disks diagram:



#include<stdio.h>

void Hanoi(int, char, char, char);

int main()

{

int n;

char Beg='A', Aux='B', End='C';

printf("Enter number of disks : ");

scanf("%d" , &n);

Hanoi(n, Beg, Aux, End);

return 0;

}

void Hanoi(int n, char Beg, char Aux, char End)

{

if(n==1)

{

printf("Move %c -> %c \n", Beg, End);

}

else

{

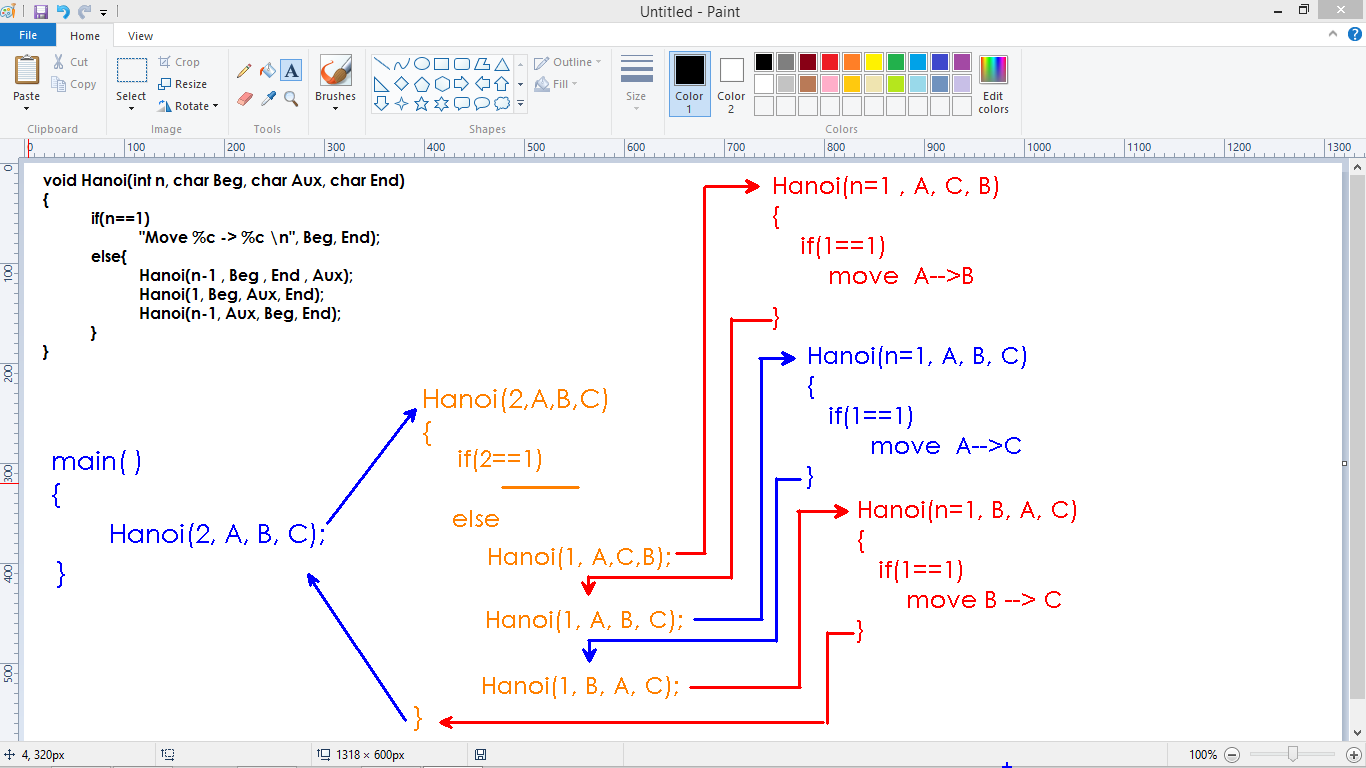
Hanoi(n-1 , Beg , End , Aux);

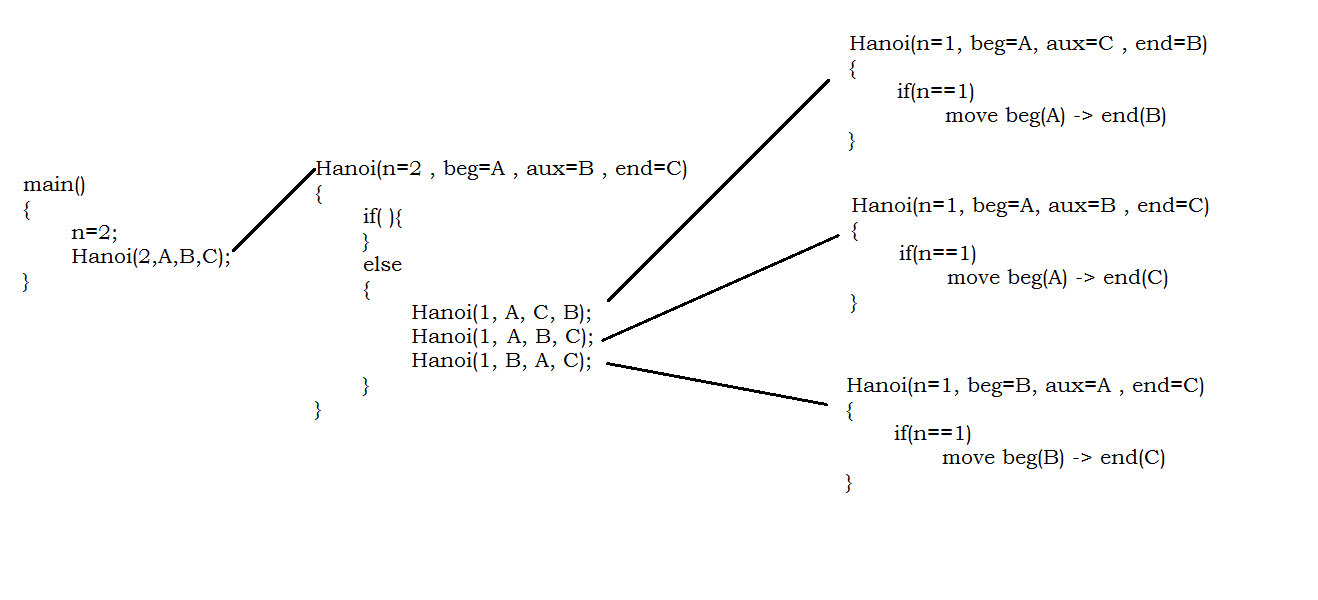
Hanoi(1, Beg, Aux, End);

Hanoi(n-1, Aux, Beg, End);

}

}





**For 3 disks:**

Enter number of disks: 3

Move A -> C

Move A -> B

Move C -> B

Move A -> C

Move B -> A

Move B -> C

Move A -> C

