

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA



SOFTWARE ENGINEERING

ASSIGNMENT-3

SUBMITTED TO: DR.MADIHA LIAQAT

SUBMITTED BY: MUHAMMAD ARSALAN

REG NO: 20-SE-56

COURSE: PF

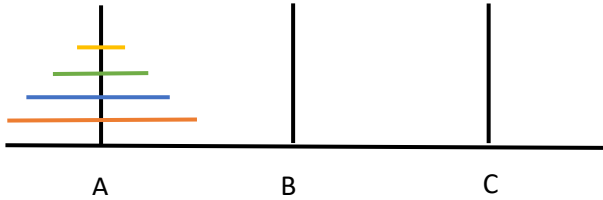
DATED: MAY8,2021

+++++

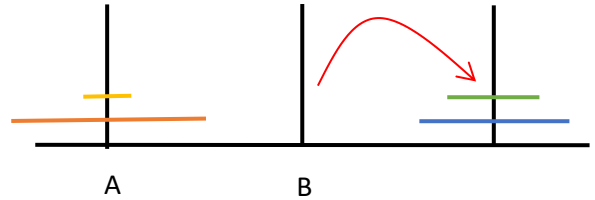
C++ PROGRAMM TO EXPLAIN TOWERS OF HANOI

Scenario user chosen 4 disc to be moved on tower named 'B'.

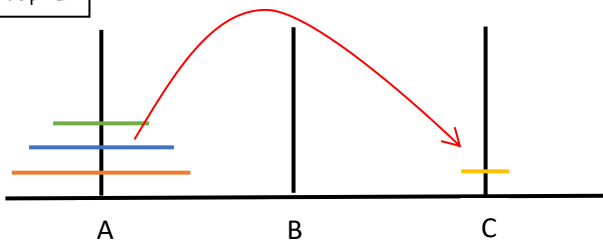
Step-1



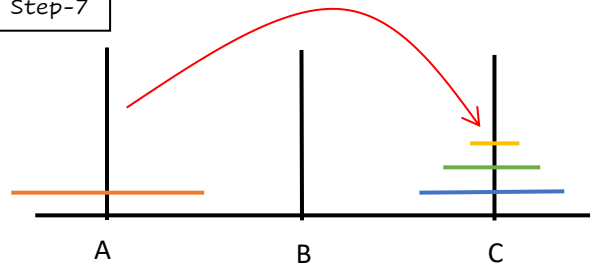
Step-6



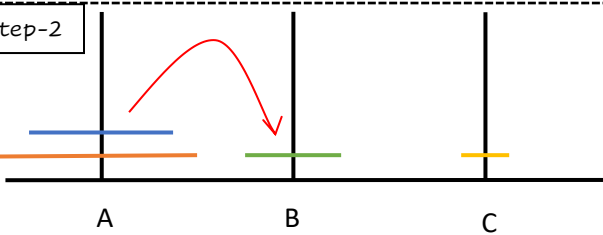
Step-1



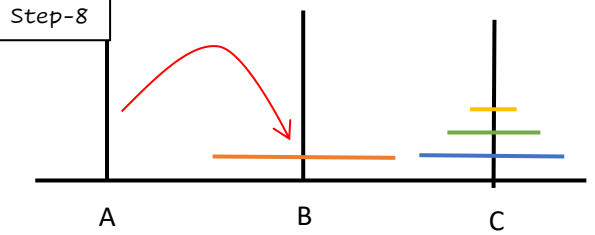
Step-7



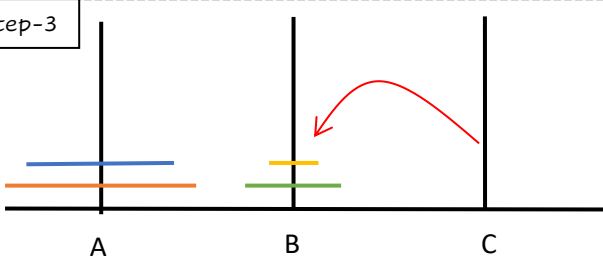
Step-2



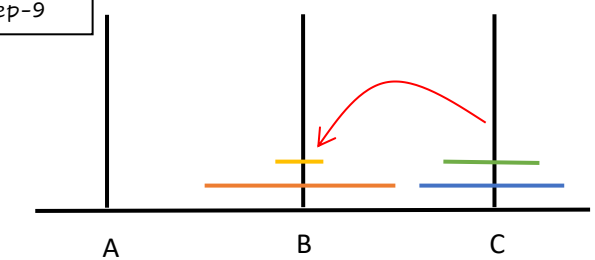
Step-8



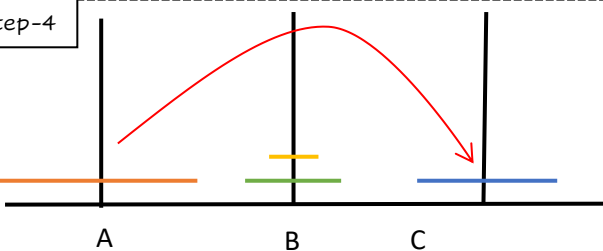
Step-3



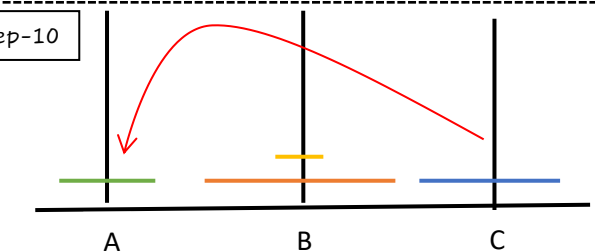
Step-9



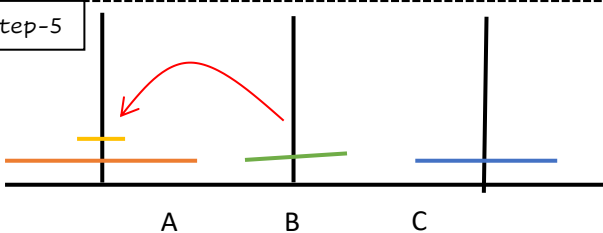
Step-4



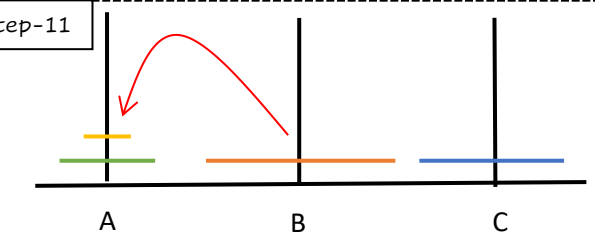
Step-10

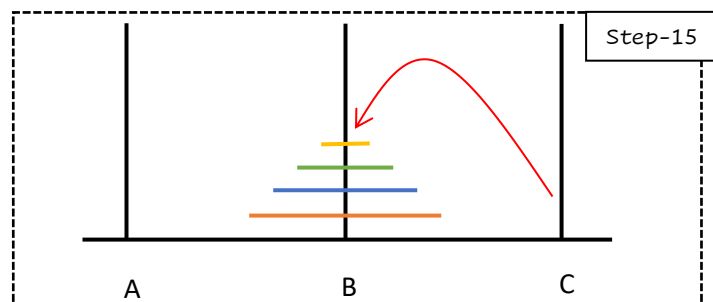
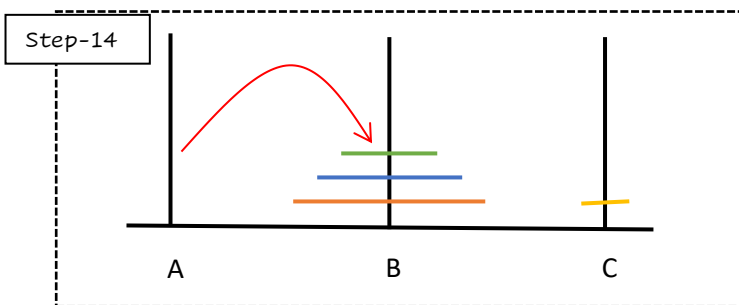
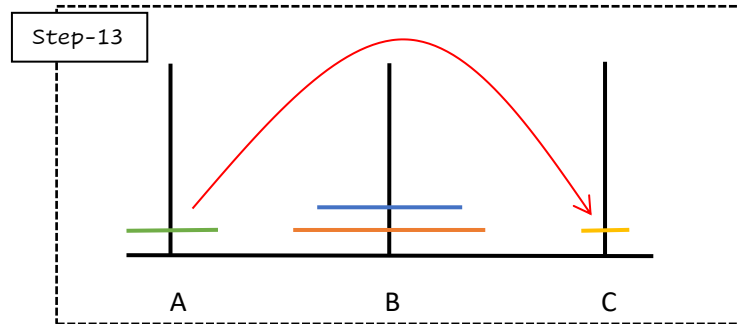
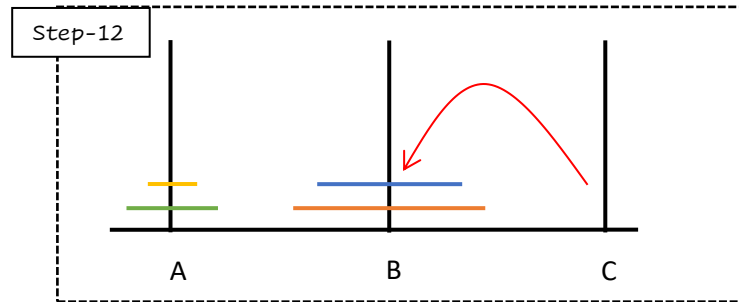


Step-5



Step-11





EXPLANATION OF WORKING

Note: above visual explanation is not plagiarised nor copied from any online source . These diagram explanation is drawn by myself for proof world document file is also submitted in the section to verify graphical explanation .in this possible scenario graphical explanation of only 4 disc is done because further greater was difficult to express and draw on such a small page.instead of containers or discs they are represented by simple lines .

1st disc=yellow

2nd disc=green

3rd disc=blue

4th disc=brown

As it is user choice to choose number of disc he want to use so we let that user choosed 4 disc to be involved in this game and there are 3 towers named as A B C .user chosed to move all disc to tower B. Game will complete in fifteen steps.

Initially at tower A we have disc

A(1,2,3,4) B() C()

In 1st step we moved disc 1 from tower A to C.

A(2,3,4) B() C(1)

In 2nd step we moved disc 2 from tower A to B.

A(3,4) B(2) C(1)

In 3rd step we moved disc 1 from tower C to B.

A(3,4) B(1,2) C()

In 4th step we moved disc 3 from tower A to C.

A(4) B(1,2) C(3)

In 5th step we moved disc 1 from tower B to A.

A(1,4) B(2) C(3)

In 6th step we moved disc 2 from tower B to C.

A(1,4) B() C(2,3)

In 7th step we moved disc 1 from tower A to C.

A(4) B() C(1,2,3)

In 8th step we moved disc 4 from tower A to B.

A() B(4) C(1,2,3)

In 9th step we moved disc 1 from tower C to B.

A() B(1,4) C(2,3)

In 10th step we moved disc 2 from tower C to A.

A(2) B(1,4) C(3)

In 11th step we moved disc 1 from tower B to A.

A(1,2) B(4) C(3)

In 12th step we moved disc 3 from tower C to B.

A(1,2) B(3,4) C()

In 13th step we moved disc 1 from tower A to C.

A(2) B(3,4) C(1)

In 14th step we moved disc 2 from tower A to B.

A() B(2,3,4) C(1)

In 15th step we moved disc 1 from tower C to B.

A() B(1,2,3,4) C()

Hence all discs are moved to tower B .

C++ CODE

// C++ PROGRAM TO MAKE TOWERS OF HANOI USING RECURSION WHERE USER CHOSSES NUMBER OF DISCS TO BE MOVED AND ON WHICH TOWER

```
#include <iostream>
```

```
using namespace std;
```

```
//DECLARED FUNCTION FOR TOWER OF HANOI
```

```
void towers_Hanoi(int,char,char,char);
```

```
// IT IS MAIN FUNCTION OF THIS PROGRAMM
```

```
int main()
```

```
{
```

```
//DEFINING VARIABLES TO BE USED IN PROGRAMM
```

```
int num_disc;
```

```
char from_tower,aux_tower,to_tower;
```

```
cout<<" Enter Number Of Disc User Wants To Use:";
```

```
cin>>num_disc;
```

```
cout<<"At Which Tower You Want To Move Disc:";
```

```
cin>>to_tower;
```

```
//HERE TERNITORY OPERATORS ARE USED IN ORDER TO CHOOSE ON WHICH  
TOWER DISCS ARE TO BE MOVED
```

```
aux_tower=(to_tower=='C'||to_tower=='c')?'B':'C';  
cout<<endl;
```

```
towers_Hanoi(num_disc, 'A', to_tower, aux_tower);  
cout<<"\n\n\nGAME HAS ENDED THANK YOU FOR PLAYING\n\n\n";
```

```
return 0;  
}  
// DEFINING FUNCTION INVOLVING DISCS MOVEMENT
```

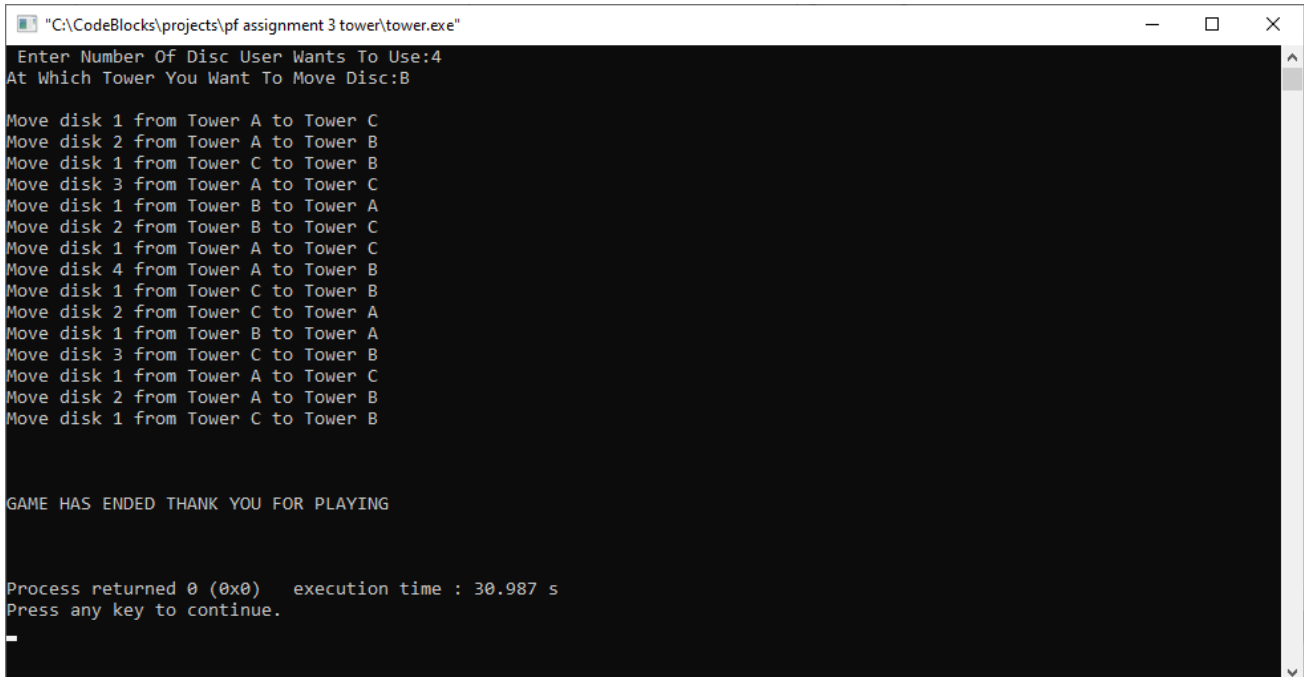
```
void towers_Hanoi(int num_disc, char from_tower,  
                  char to_tower, char aux_tower)  
{  
    if (num_disc== 1)  
    {  
        cout << "Move disk 1 from Tower " << from_tower <<  
            " to Tower " << to_tower<<endl;  
        return;  
    }  
    else
```

```
//HERE RECURSION IS INVOLVED FOR USING CALLING SAME TOWER OF HANOI  
FUNCTION AGAIN AND AGAIN
```

```
{  
  
    towers_Hanoi(num_disc - 1, from_tower, aux_tower, to_tower);  
    cout << "Move disk " << num_disc << " from Tower " << from_tower <<  
        " to Tower " << to_tower << endl;  
    towers_Hanoi(num_disc - 1, aux_tower, to_tower, from_tower);  
}
```

}

OUTPUT



```
"C:\CodeBlocks\projects\pf assignment 3 tower\tower.exe"
Enter Number Of Disc User Wants To Use:4
At Which Tower You Want To Move Disc:B

Move disk 1 from Tower A to Tower C
Move disk 2 from Tower A to Tower B
Move disk 1 from Tower C to Tower B
Move disk 3 from Tower A to Tower C
Move disk 1 from Tower B to Tower A
Move disk 2 from Tower B to Tower C
Move disk 1 from Tower A to Tower C
Move disk 4 from Tower A to Tower B
Move disk 1 from Tower C to Tower B
Move disk 2 from Tower C to Tower A
Move disk 1 from Tower B to Tower A
Move disk 3 from Tower C to Tower B
Move disk 1 from Tower A to Tower C
Move disk 2 from Tower A to Tower B
Move disk 1 from Tower C to Tower B

GAME HAS ENDED THANK YOU FOR PLAYING

Process returned 0 (0x0)   execution time : 30.987 s
Press any key to continue.
_
```

EXPLANATION OF PROGRAM WORKING

- After writing header files we first declared a function named "towers_hanoi" and in its scope we initialized one integer and three character type variables
- After initializing this function we started our main body in which we declared an integer type variable named "num_disc" and three character type variables named as "from_tower", "aux_tower", "to_tower"
- Then we asked user to enter number of disc he wants to use in this game and this is stored in integer variable "num_disc"
- After this user is also asked to choose on which tower he wants to move these discs and this choice is stored in the form of character, in our program either B or C and that is stored in character type variable named as "to_tower"
- In next step we have used ternary operator to process user choice for on which tower he wants to move discs,
- Here purpose of auxiliary tower/rod is it helps in movement of disks.
- In next line of code in function call line we have replaced variable named "from_tower" to "A" because we are aware that initially all disks are at tower A.
- And here main function body is ended then we define our function that we declared earlier in the starting of program.

- We used if else statement for working of these disks movement. firstly in recursive case I used the method two times so that we print the message again.
- In last phase of “else” statements recursion is ready to be used in main function .in this we assign different towers value and take the input of number of rings and here due to recursion this function calls itself again and again and hence puzzle is solved.