

source files: stack.h, stack.c

Main:

```
//While loop to get the input from the user
    //Switch to mark the input option
        Case1: get the number of Hanoi
        Case2: mark Stack is true
        Case3: mark Recursion is true
//If statement to run function to calculate the Hanoi process
    If Stack is true
        call Stack function
    End if
    If Recursion is true
        call Recursion function
    End if
```

Function for recursive program Recursion(number of disk, left, middle, right)

```
If disk == 0, THEN:
    move disk from left to middle
Else:
    Recursion(number of disk - 1, left, right, middle)
    move disk from left to middle
    Recursion(number of disk - 1, left, middle, right)
End if
```

Function for stack program(number of disks)

playWithStack(int nDisks)

 print prompt string

 initialize pega (push disks)

 create pegb, pegc

 arrange the three pegs as a triangle according to nDisks (odd/even)

 while(sum of move < $2^n - 1$)

 step 1. move disk 1 to the peg next to it

 step 2. move between the other two pegs

 End while

 release the space of stacks

 return

step 1.

 if disk 1 is on pega

 pega -> pegb

 elseif disk 1 is on pegb

 pegb -> pegc

 else

 pegc -> pega

step 2.

 if there is one peg empty

 move the disk from the peg where disk1 is not on to the empty peg

 End if

 Else

 find the two pegs that don't have disk1,

 move the smaller number disk to the peg that has larger number

 End else

 source files: stack.h

//define functions declared in stack.h