1BM17CS050

N. Jayanth

ADA Lab Test

14/6/21

16CS5BCADA

1) Write a recursive function

a) to find GCD

```
#include <stdlib.h>
#include <stdio.h>
int gcd (int a, int b)
int main( )
{
  int a, b;
printf ("Enter two integers");
scanf ("%d %d", &a, &b);
printf ("GCD of %d & %d is %d", a, b, gcd (a,b));
return 0;
int gcd (int a, int b)
{
  if (b != 0)
   return gcd (b, a%b);
  else
   return a;
}
```

b) original program to find towers of Hanoi using recursive function.

```
b) #include <stdlib.h>
   #include <stdio.h>
   void hanoi (int n, char x, char y, char z);
   {
     if (n == 1)
     { printf ("Move disk 1 from rod %d to rod %d", &x, &y);
       return;
     }
     hanoi ( n-1, x, z, y);
     printf ("move disk %d from rod %d to rod %d", &n, &x, &y);
     hanoi ( n-1, z, y, x);
   }
   void main ()
   { int n;
     printf ("Enter number of disks ");
     scanf ("%d", &n);
     hanoi ( n, "a", "c", "b");
   }
```

b) modified program count number of recursive calling in the tower of Hanoi problem.

```
#include <stdlib.h>
#include <stdio.h>
void hanoi (int n, char x, char y, char z, int count)
{ if (n == 1)
  { count++;
    return;
  }
```

```
void hanoi( int n, char x, char y, char z, int count)
{
    if (n == 1)
        printf("Error");
    else
        hanoi(n-1, x, z, y)
        cout << "Move disk" << n << "from rod" << x << "to rod" << y;
        hanoi(n-1, z, y, x)

void hanoi ( int n, char x, char y, char z)
{   int count;
    if ( n == 1)
        printf(" Move disk %d from rod %c to rod %c", n, x, y);
        return ;
    else
        count= hanoi(n-1, x, y, z, count);
        printf(" The count of recursive call for %d ", count);
        count= hanoi(n-1, x, y, z, count);
        printf(" Move disk %d from rod %c to rod %c", n, x, y);
        count= hanoi (n-1, z, y, x);
        printf ("The count of recursive calls = %d ", count);
    return count;
}

void main()
{
    int n,h;
    printf ("Enter number of disks");
    scanf("%d", &n);
    h= hanoi( n, 'a', 'c', 'b');
    printf ("The number of recursive calls are %d", h);
}
```