**PROGRAM 1**

**WRITE A RECURSIVE PROGRAM TO :**

**A)SOLVE TOWERS-OF-HANOI PROBLEM**

**B)TO FIND GCD**

**Recursive program to find GCD:**

#include <stdio.h>

#include <time.h>

int gcd(int a, int b)

{

if (b == 0)

return a;

return gcd(b, a % b);

}

int main()

{

int a,b;

clock\_t start, end;

double time;

printf("Greatest common divisor\n");

printf("Enter two numbers\n");

scanf("%d%d",&a,&b);

start = clock();

printf("GCD of %d and %d is %d \n", a, b, gcd(a, b));

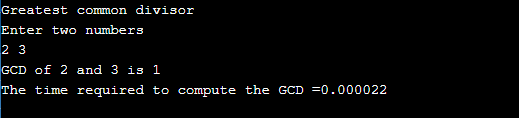
end = clock();

time= ((double) (end - start)) / CLOCKS\_PER\_SEC;

printf("The time required to compute the GCD =%f",time);

}

Output:



**Recursive program to solve Tower-of-Hanoi Problem**

#include <stdio.h>

#include <time.h>

void TOH(int, char, char, char);

int main()

{

int num;

clock\_t start, end;

double time;

printf("TOWER OF HAN0I PROBLEM\n");

printf("Enter the number of disks : \n");

scanf("%d", &num);

printf("The sequence of moves involved in the Tower of Hanoi are :\n");

start=clock();

TOH(num, 'A', 'C', 'B');

end = clock();

time= ((double) (end - start)) / CLOCKS\_PER\_SEC;

printf("\n\nThe time required to compute the Tower of Hanoi problem is =%f",time);

}

void TOH(int num, char frompeg, char topeg, char auxpeg)

{

if (num == 1)

{

printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg);

return;

}

TOH(num - 1, frompeg, auxpeg, topeg);

printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);

TOH(num - 1, auxpeg, topeg, frompeg);

}

**Output:**

