

i) Solve tower of Hanoi.

```
#include <stdio.h>

void towers (int, char, char, char);

int main ()
{
    int num;
    printf ("Enter the number of disks:");
    scanf ("%d", &num);
    printf ("The sequence of moves involved in the tower of hanoi are:\n");
    towers (num, 'A', 'C', 'B');
    return 0;
}

void towers (int num, char a, char b, char c)
{
    if (num == 1)
    {
        printf ("\n Move disk 1 from peg %c to peg %c", a, b);
        return;
    }
    towers (num-1, a, b, c);
    printf ("\n Move disk %d from peg %c to peg %c", num, a, b);
    towers (num-1, c, b, a);
}
```

GCD:

```

#include <stdio.h>
int gcd (int m, int n)
{
    if (n == 0)
        return m;
    if (m < n) return
        return gcd(n, m);
    return gcd(n, m % n);
}
int main()
{
    int m, n, res;
    printf ("Enter m and n\n");
    scanf scanf ("%d %d", &m, &n);
    res = gcd
    res = gcd (m, n);
    printf ("GCD of %d and %d is %d", m, n, res);
}

```

Count - the number of recursive call-in solving tower-of-hanoi problem.

```

int clock = t, end, start;
printf ("1n");
start = clock();
for (int i = 0; i < last - 1; i++)
{
    int j = i + 1;
    for (; j < last; j++)
    {
        if (arr[j] < arr[i]) {
            arr[i] += arr[j]

```

(2)

S.R. Podja


```
arr[j] = arr[i] - arr[j];  
arr[i] = arr[i] - arr[j];  
}
```

```
} end = clock();
```

```
printf("Time in sec : %f\n", ((double)(end - start)) / CLOCK_PER_SEC);  
for (int i = 0; i < last; i++)  
    printf("%d\t", arr[i]);  
}
```