~~LAB~~ PRACTISE PROGRAMS:→ Tower of Hanoi

```
#include <stdio.h>
void towers (int n, char src, char temp, char dest)
{
  // if only one disk moves and returns
  if (n==1)
  {
     printf("Move disk %d from %c to %c", n, src, dest);
     return;
  }
//move n-1 disks from src to temp using dest peg
towers (n-1, src, dest, temp);
//move remaining disks from src to dest peg
printf("\n Move disk %d from %c to %c \n", n, src, dest);
//move n-1 disks from temp to dest using source peg
towers (n-1, temp, src, dest);
}
int main()
 {
   int n;
   printf("Enter the number of disks \n");
   scanf("%d", &n);
   towers(n, 'S', 'T', 'D');
 }
```

OUTPUT:

```
Enter the number of disks
2
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
```

Scanned with CamScanner

```
Enter the number of disks
4
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
Move disk 3 from S to T
Move disk 1 from D to S
Move disk 2 from D to T
Move disk 1 from S to T
Move disk 4 from S to D
Move disk 1 from T to D
Move disk 2 from T to S
Move disk 1 from D to S
Move disk 3 from T to D
Move disk 1 from S to T
Move disk 2 from S to D
Move disk 1 from T to D
[Program finished]
```

Scanned with CamScanner

→ Fibonacci series of n numbers.

```
#include <stdio.h>
int fib(int n)
{
  if (n==0) return 0;
  if (n==1) return 1;
  return fib(n-1)+fib(n-2);
}
int main()
{
  int i,n;
  printf("Enter n\n");
  scanf("%d",&n);
  printf("%d fibonacci numbers are:\n",n);
  for(i=0;i<n;i++)
  {
    printf("fib(%d)=%d\n",i,fib(i));
  }
}
```

OUTPUT:

```
Enter n
7
7 fibonacci numbers are:
fib(0)=0
fib(1)=1
fib(2)=1
fib(3)=2
fib(4)=3
fib(5)=5
fib(6)=8
```

Scanned with CamScanner

```
Enter n
9
9 fibonacci numbers are:
fib(0)=0
fib(1)=1
fib(2)=1
fib(3)=2
fib(4)=3
fib(5)=5
fib(6)=8
fib(7)=13
fib(8)=21

[Program finished]
```

Scanned with CamScanner

→ Factorial of n numbers:

```
#include <stdio.h>
int fact (int n)
{
  if (n==0) return 1;
  return  n * fact (n-1) ;
}
int main()
 {
   int n;
   printf ("Enter the value of n \n");
   scanf ("%d", &n);
   printf ("The factorial of %d = %d \n", n, fact(n));
 }
```

OUTPUT:

```
Enter the value of n
5
The factorial of 5 = 120
```

Scanned with CamScanner

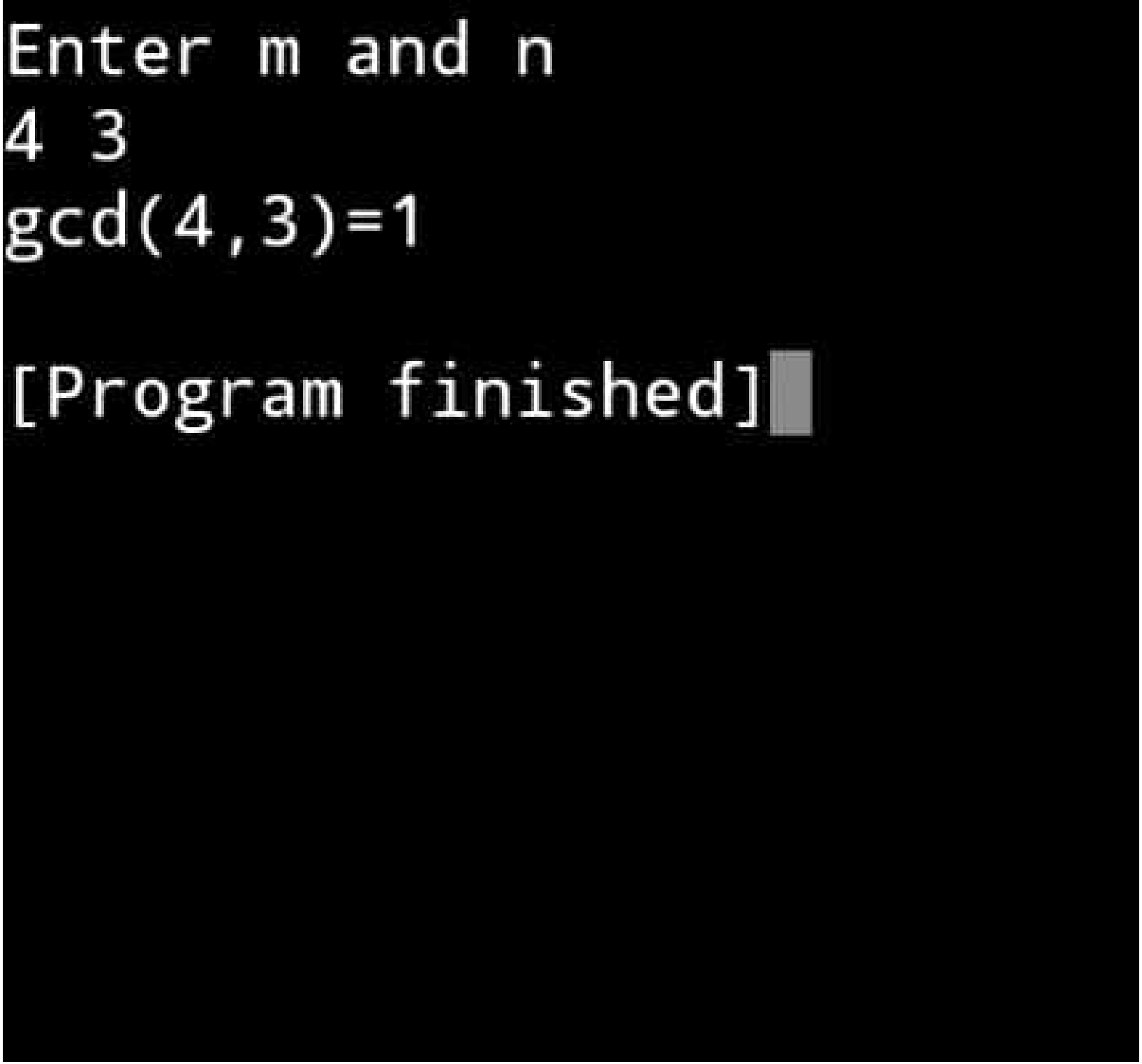
```
Enter n
5
The factorial of 5=120

[Program finished]
```

Scanned with CamScanner

→ GCD of two numbers.

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

Scanned with CamScanner



Scanned with CamScanner

→ Binary search using recursion.

```
#include <stdio.h>
void binary_search(int [], int, int, int);
void bubble_sort(int [], int);
int main()
{
 int key, size, i;
 int list [25];
 printf("Enter size of a list : ");
 scanf("%d", &size);
 printf("Enter elements \n");
 for(i=0; i<size; i++)
  {
   scanf("%d", &list[i]);
  }
 bubble_sort(list, size);
 printf("\n");
 printf("Enter key to search \n");
 scanf("%d", &key);
 binary_search(list, 0, size, key);
}
void bubble_sort(int list [], int size)
{
 int temp, i, j;
 for(i=0; i< size; i++)
  {
   for(j=i; j< size; j++)
    {
     if(list[i] > list[j])
      {
       temp = list[i];
       list[i] = list[j];
```

Scanned with CamScanner

```
        list[j] = temp;
      }
    }
  }
}
void binary_search (int list[], int lo, int hi, int key)
{
  int mid;
  if (lo > hi)
  {
    printf ("Key not found \n");
    return;
  }
  mid = (lo + hi)/2;
  if (list[mid] == key)
  {
    printf ("Key found \n");
  }
  else if (list[mid] > key)
  {
    binary_search (list, lo, mid-1, key);
  }
  else if (list[mid] < key)
  {
    binary_search (list, mid+1, hi, key);
  }
}
```

OUTPUT:

```
Enter size of a list: 4
Enter elements
1 2 3 4
Enter key to search
3
```

→ Key found.

Scanned with CamScanner

```
Enter size of a list: 4
Enter elements
1 2 3 4

Enter key to search
3
Key found

[Program finished]
```

Scanned with CamScanner