// program 01 TOWER OF HENNOI

```
#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, 'S', 'T', 'D');
  return 0;
}
void towers(int num, char SRC, char TEMP, char DES)
{
  if (num == 1)
  {
    printf("\n Move disk 1 from peg %c to peg %c", SRC, DES);
    return;
  }
  towers(num - 1, SRC, TEMP, DES);
  printf("\n Move disk %d from peg %c to peg %c", num, SRC, DES);
  towers(num - 1, DES, SRC, TEMP);
}
```

program 2   FACTORIAL

```
#include <stdio.h>
int fact (int);
int main()
{
   int n,f;
   printf("Enter the number whose factorial you want to calculate?");
   scanf("%d",&n);
   f = fact(n);
   printf("factorial = %d",f);
}
int fact(int n)
{
   if (n==0)
   {
      return 1;
   }
   else if ( n == 1)
   {
      return 1;
   }
   else
   {
      return n*fact(n-1);
   }
}
```

//PROGRAM 3 FIBONNACI SERIES:

```
#include<stdio.h>

int fibonacci(int);

void main ()

{

   int n,f;

   printf("Enter the value of n\n");

   scanf("%d",&n);

   f = fibonacci(n);

   printf("%d",f);

}

int fibonacci (int n)

{

   if (n==0)

   {

   return 0;

   }

   else if (n == 1)

   {

       return 1;

   }

   else

   {

       return fibonacci(n-1)+fibonacci(n-2);

   }

}
```

//PROGRAM 04 FIND GCD:

```
#include <stdio.h>

int hcf(int n1, int n2);

int main() {

    int n1, n2;

    printf("Enter two positive integers: ");

    scanf("%d %d", &n1, &n2);

    printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));

    return 0;

}


int hcf(int n1, int n2) {

    if (n2 != 0)

        return hcf(n2, n1 % n2);

    else

        return n1;

}
```

//PROGRAM 05 Implement recursive Binary Search

```
#include <stdio.h>

int binarySearch(int arr[], int l, int r, int x)

{


   if (r >= l) {

       int mid = l + (r - l) / 2;

       if (arr[mid] == x)

           return mid;


       if (arr[mid] > x)

           return binarySearch(arr, l, mid - 1, x);

       return binarySearch(arr, mid + 1, r, x);

   }

   return -1;

}


int main(void)

{

   int arr[] = { 2, 3, 4, 10, 40 };

   int n = sizeof(arr) / sizeof(arr[0]);

   int x = 10;

   int result = binarySearch(arr, 0, n - 1, x);

   (result == -1) ? printf("Element is not present in array") : printf("Element is present at index %d", result);


   return 0;

}
```