

```
541 //Factorial using Recursion
542 #include<stdio.h>
543 unsigned long long int calc(int n)
544 {
545     if(n==0)
546         return(1);
547     return(n*calc(n-1));
548 }
549 int main()
550 {
551     int n;
552     unsigned long long int f;
553     printf("Enter a number whose factorial you want to find : ");
554     scanf("%d",&n);
555     f=calc(n);
556     printf("Factorial of %d is : %llu",n,f);
557     return 0;
558 }
```



"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"


Enter a number whose factorial you want to find : 17

Factorial of 17 is : 355687428096000

Process returned 0 (0x0) execution time : 3.399 s

Press any key to continue.

```
560
561 //GCD using Recursion
562 #include <stdio.h>
563 int hcf(int n1, int n2)
564 {
565     if (n2 != 0)
566         return hcf(n2, n1%n2);
567     else
568         return n1;
569 }
570 int main() {
571     int n1, n2;
572     printf("Enter two positive integers : \n");
573     scanf("%d%d", &n1, &n2);
574     printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
575     return 0;
576 }
577
```

 "C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

Enter two positive integers :

13

91

G.C.D of 13 and 91 is 13.

Process returned 0 (0x0) execution time : 5.311 s

Press any key to continue.

```
579 //Fibonacci using recursion
580 #include<stdio.h>
581 int f(int n){
582     if(n==0)
583         return 0;
584     else if(n==1)
585         return 1;
586     else
587         return f(n-1)+f(n-2);
588 }
589 int main(){
590     int i,n,m=0;
591     printf("Enter n : ");
592     scanf("%d",&n);
593     printf("%d fibonacci series is : \n",n);
594     for(i=1;i<=n;i++){
595         printf("f(%d)=%d\n",i,f(m));
596         m++;
597     }
598     return 0;
599 }
600
```



"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

Enter n : 5

5 fibonacci series is :

f(1)=0

f(2)=1

f(3)=1

f(4)=2

f(5)=3

Process returned 0 (0x0) execution time : 3.203 s

Press any key to continue.

```

602 //Binary Search using recursion
603 #include<stdio.h>
604 int Search(int n, int a[n], int s, int e, int k)
605 {
606     if(e>=s){
607         int m=s+(e-s)/2;
608         if(a[m]==k){
609             return m;
610         }
611         else if(a[m]>k){
612             return Search(n,a,s,m-1,k);
613         }
614         return Search(n,a,m+1,e,k);
615     }
616     return -1;
617 }
618 int main()
619 {
620     int n,i,x,g;
621     printf("Enter the number of elements in array : ");
622     scanf("%d",&n);
623     int a[n];
624     printf("Enter elements of array : \n");
625     for(i=0;i<n;i++){
626         scanf("%d",&a[i]);
627     }
628     printf("Enter the element to be searched : ");
629     scanf("%d",&g);
630     x=Search(n,a,0,n-1,g);
631     if(x== -1){
632         printf("Element not found in the array.");
633     }
634     else{
635         printf("Element is present in array and its position is : %d",x+1);
636     }
637     return 0;
638 }
639

```



"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

Enter the number of elements in array : 4

Enter elements of array :

12

43

56

89

Enter the element to be searched : 56

Element is present in array and its position is : 3

Process returned 0 (0x0) execution time : 9.838 s

Press any key to continue.


```
641 //Tower of Hanoi
642 #include<stdio.h>
643 void towers(int n, char s, char t, char d)
644 {
645     if (n==1) {
646         printf("Move disk 1 from %c to %c\n", s, d);
647         return;
648     }
649     towers(n-1, s, d, t);
650     printf("Move disk %d from %c to %c\n", n, s, d);
651     towers(n-1, t, s, d);
652 }
653 int main()
654 {
655     int n;
656     printf("Enter no. of disks : ");
657     scanf("%d", &n);
658     towers(n, 'S', 'T', 'D');
659     return 0;
660 }
661
```



"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

Enter no. of disks : 3

Move disk 1 from S to D

Move disk 2 from S to T

Move disk 1 from D to T

Move disk 3 from S to D

Move disk 1 from T to S

Move disk 2 from T to D

Move disk 1 from S to D

Process returned 0 (0x0) execution time : 1.351 s

Press any key to continue.