

Recursion : Factorial of a no.

```
#include <stdio.h>
```

```
int factorial(int n);
```

```
int main()
```

```
{ int n;
```

```
    printf("Enter a no.");
```

```
    scanf("%d", &n);
```

```
    printf("Factorial of %d is %d", n, factori  
al(n));
```

```
    return 0;
```

```
}
```

```
int factorial(int n) {
```

```
    if (n >= 1)
```

```
        return n * factorial(n-1);
```

```
    else
```

```
        return 1;
```

```
}
```

GCD of 2 nos. using recursion.

```
#include <stdio.h>
```

```
int hcf(int n1, int n2)
```

```
int main()
```

```
{ int n1, n2;
```

```
    printf("Enter 2 integers");
```

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

```
    printf("GCD of %d and %d is %d", n1, n2,  
hcf(n1, n2));
```

```
    return 0;
```

```
}
```

```
int hcf(int m1, int n2) {
    if (m2 == 0)
        return hcf(m2, m1 % m2);
}
```

```
else
    return 1;
}
```

Tower of hanoi using recursion.

#include <stdio.h>

```
void towerOfHanoi(int n, char from from,
    char to, char aux) {
    if (n == 1)
    { printf("Move disc 1 from %c to
        to %c", from, to);
        return;
    }
}
```

```
towerOfHanoi(n-1, from, to, aux);
printf("Move disc %d from %c to
    %c", n, from, to);
towerOfHanoi(n-1, aux, to, from);
}
```

```
int main()
{ int n = 4;
    towerOfHanoi(n, 'A', 'C', 'B');
    return 0;
}
```

Que's simplification



## Binary Search-

```
#include <stdio.h>
```

```
void binary_search (int l[], int, int) {
```

```
void bubble_sort (int l[], int);
```

```
int main()
```

```
{ int key, size, i;
```

```
int list[25];
```

```
printf("Enter size of list");
```

```
scanf("%d", &size);
```

```
printf("Enter Elements. \n");
```

```
for (i=0; i < size; i++)
```

```
{ scanf("%d", &list[i]); }
```

```
bubble_sort (list, size);
```

```
printf("Enter key to search ");
```

```
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+1; j < size; j++)
```

```
{ if (list[i] > list[j])
```

```
{ temp = list[j];
```

```
list[i] = list[j];
```

```
list[j] = temp;
```

```
}}}}
```

```
void binary_search (int list[], int low, int high, int key)
```

```
{ int mid;
```

```
if (low > high)
```

```
{ printf("key not found");
```

return ; }  
mid = (low + high) / 2;  
if (list[mid] == key)  
{ printf("Key found"); }  
else if (list[mid] < key)  
{ binary-search(list, mid + 1, high, key);  
}  
}

Date

Page

x Terminal



Enter size of a list: 3

Enter elements

22 55 78

Enter key to search

55

Key found

Process finished.





 Terminal

```
Enter a positive integer: 8  
Factorial of 8 = 40320  
Process finished.
```



x Terminal



Enter n

7

fib nos 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

Process finished.



x Terminal



enter the value of n2

Move disk 1 from rod A to rod B  
Move disk 2 from rod A to rod C  
Move disk 1 from rod B to rod C  
Process finished.





x Terminal



Enter two positive integers: 64 44  
G.C.D of 64 and 44 is 4.  
Process finished.

