

① Binary Search using Recursion

```
#include <stdio.h>
int binarySearch(int arr[], int l, int h, int val);
int n;
int main() {
    int arr[100], val;
    printf("Enter the number of elements\n");
    scanf("%d", &n);
    printf("Enter the elements of array\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter the element to be searched\n");
    scanf("%d", &val);
    int answer = binarySearch(arr, 0, n - 1, val);
    if (answer == -1)
        printf("Element is not present\n");
    else
        printf("Element is present at %d\n", answer);
}
```

Date ___ / ___ / ___

```
int binarysearch(int a[], int l, int h,  
                 int item)
```

{

```
    int mid = (l + h) / 2
```

```
    if (a[mid] == item)  
        return mid;
```

```
    else if (a[mid] > item)  
        return binarysearch(a, l, mid - 1,  
                            item);
```

```
    else if (a[mid] < item)  
        return binarysearch(a, mid + 1, h,  
                            item);
```

else

```
    return -1;
```

output of Binary Search

Enter the number of elements of array

5

Enter the elements of array

8

10

9

~~5~~ 5

4

*** Array after sorting is ***

4

5

8

9

10

Enter the element to be searched

8

Element is present at location 2

② Factorial using Recursion

```
#include <stdio.h>
int fact(int n);
int main()
{
    int n, val;
    printf("Enter the number n: ");
    scanf("%d", &n);
    val = fact(n);
    printf("Factorial is %d\n", val);
```

Saath

Date _____

```
int fact(int n)
{
    if (n == 1)
        return 1;
    else
        return n * fact(n-1);
```

Date ___ / ___ / ___

output of factorial

Enter the number whose factorial is to be found

5

Factorial of number is 120

③

Fibonacci using recursion

* `#include < stdio.h >`

`int fib (int n);`

`int main ()`

`{`

`int n, Val;`

`printf ("Enter the number of terms
n");`

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

`printf ("Fibonacci sequence is %d\n");`

`for (int i=0; i<n; i++)`

`{`

`Val = fib (i);`

`printf ("%d\t", Val);`

`return 0;`

`}`

Date ___ / ___ / ___

```
int fib(int n)
{
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return (fib(n-1) + fib(n-2));
}
```

output of fibonacci

Enter the number of terms

5

Fibonacci numbers of 5 is

0

1

1

2

3

① GCD using recursion

#include < stdio.h >

int gcd (int x, int y)

int main()

{

int x, y;

printf ("Enter two numbers\n");

scanf ("%d %d", &x, &y);

int answer = gcd(x, y);

printf ("GCD is: %d\n", answer);

}

int gcd (int x, int y)

{

int rem = x % y

if (rem == 0)

return y;

else

return gcd(y, rem);

}

output of GCD

Enter two numbers

10
4

GCD of 10 and 4 is 2

(3) Tower of hanoi

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

```
void move (int n, char a, char b, char c);
```

```
int main ()
```

```
{
```

```
int n;
```

```
printf ("Enter the number of discs to be moved ");
```

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

```
printf ("A: source B: spare C: Destination");
```

```
move (n, 'A', 'C', 'B');
```

```
return 0;
```

```
Void move (int n, char source, char destination, char spare)
```

```
{
```

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

```
printf ("Move from %c to %c\n", source, destination);
```

```
else
```

```
{
```

```
move (n-1, source, spare, destination);
```

```
move (1, source, destination, spare);
```

Saathi

Date ___ / ___ / ___

move (n-1, spare, destination, tail);

Date _____ / _____ / _____

Output for Tower of Hanoi

Enter the number of discs

3

A : Source

B : spare

C : Destination

Move from A to C

Move from A to B

Move from C to B

Move from A to C

move from B to A

move from B to C

move from A to C