Binary Search:

```c
#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()
{
    int i, n, x;
    printf("Enter array size\n");
    scanf("%d", &n);
    int a[n];
    for (i = 0; i < n; i++){
    printf("Enter the element: ");
    scanf("%d", &a[i]);
    }
    printf("Enter the number to be searched\n");
    scanf("%d", &x);
    int m = binarysearch(a, 0, n - 1, x);
    if (m == -1)
        printf("Search unsuccessful\n");
    else
        printf("Element present at index: %d\n", m);
    return 0;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                  2: Code
PS D:\Prg data> cd "d:\Prg data\C&C++\Stack implementation\" ; if ($?) { gcc binary_search.c -o binary_search } ; if ($?) { .\binary_search }
Enter array size
5
Enter the element: 10
Enter the element: 15
Enter the element: 17
Enter the element: 24
Enter the element: 30
Enter the number to be searched
24
Element present at index: 3
PS D:\Prg data\C&C++\Stack implementation>
```

Tower of Hanoi:

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

void towerofhanoi(int n, char from_rod, char to_rod, char temp_rod){
    if (n==1)
    {
        printf("\nMove disk 1 from %c to rod %c", from_rod, to_rod);
        return;
    }
    towerofhanoi(n-1, from_rod, temp_rod, to_rod);
    printf("\nMove disk %d from rod %c to rod %c",n, from_rod, to_rod);
    towerofhanoi(n-1,temp_rod,to_rod,from_rod);
}

int main(){
    int n;
    printf("Enter number of disks\n");
    scanf("%d",&n);
    towerofhanoi(n,'A','C','B');
    return 0;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                  2: Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Prg data> cd "d:\Prg data\C&C++\Stack implementation\" ; if ($?) { gcc tower_of_hanoi.c -o tower_of_hanoi } ; if ($?) { .\tower_of_hanoi }
Enter number of disks
3

Move disk 1 from rod A to rod C
Move disk 2 from rod A to rod B
Move disk 1 from rod C to rod B
Move disk 3 from rod A to rod C
Move disk 1 from rod B to rod A
Move disk 2 from rod B to rod C
Move disk 1 from rod A to rod C
PS D:\Prg data\C&C++\Stack implementation>
```

GCD of two numbers:

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

int gcd(int x, int y){
    if(y!=0)
    return gcd(y, x%y);
    else
    return x;
}
int main(){
    int m,n;
```

```c
    printf("Enter numbers whose GCD is to be found: ");
    scanf("%d%d",&m,&n);
    printf("GCD of %d and %d is %d\n",m,n,gcd(m,n));
    return 0;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Prg data> cd "d:\Prg data\C&C++\Stack implementation\" ; if ($?) { gcc gcd.c -o gcd } ; if ($?) { .\gcd }
Enter numbers whose GCD is to be found: 81 45
GCD of 81 and 45 is 9
PS D:\Prg data\C&C++\Stack implementation>
```

Factorial of a number:

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

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

int main(){
    int x;
    printf("Enter the number: ");
    scanf("%d",&x);
    printf("Factorial of %d is % d\n",x,fact(x));
    return 0;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Prg data> cd "d:\Prg data\C&C++\Stack implementation\" ; if ($?) { gcc factorial.c -o factorial } ; if ($?) { .\factorial }
Enter the number: 6
Factorial of 6 is  720
PS D:\Prg data\C&C++\Stack implementation>
```

Nth Fibonacci term:

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

int fib(int n){
    if (n<=1) return n;
    return fib(n-1)+fib(n-2);
}
int main(){
    int x;
    printf("Enter number of terms: ");
    scanf("%d",&x);
    printf("%d",fib(x));
    return 0;
}
```

Output: