1. Write a program to print fibonacci Series using Recursion

```cpp
#include <iostream>
using namespace std;
int printFib(int x) {
   if((x==1)||(x==0)) {
      return(x);
   }else {
      return(printFib(x-1)+printFib(x-2));
   }
}
int main() {
   int x , i=0;
   cout << "Enter the number of terms of series : ";
   cin >> x;
   cout << "\nFibonnaci Series : ";
   while(i < x) {
      cout << " " << printFib(i);
      i++;
   }
   return 0;
}
```

Output:

```
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 % cd "/Users/supreethkumarjagarlamudi/Documents/CSA06/" && g++ fibonacciUsingRecursion.cpp -o fi
bonacciUsingRecursion && "/Users/supreethkumarjagarlamudi/Documents/CSA06/"fibonacciUsingRecursion
Enter the number of terms of series : 10

Fibonnaci Series :  0 1 1 2 3 5 8 13 21 34%
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 %
```

Time Complexity:

Fibonacci series program is O(2^n)

Space Complexity:

Fibonacci series program is O(n)

2. Write a program to check the given number is Armstrong or not using recursive function

```cpp
#include<iostream>
#include<cmath>
using namespace std;
int check_ArmstrongNumber(int num)
{
     if(num>0)
    return (pow(num%10,3) +check_ArmstrongNumber(num/10));
}
int main()
{
    int num;
    cout<<"Enter a number:";
    cin>>num;
   if(check_ArmstrongNumber(num)==num)
    cout<<"It is an Armstrong Number";
   else
    cout<<"It is not an Armstrong Number";
}
```

Output:

```
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 % cd "/Users/supreethkumarjagarlamudi/Documents/CSA06/" && g++ armstrongUsingRecursion.cpp -o ar
mstrongUsingRecursion && "/Users/supreethkumarjagarlamudi/Documents/CSA06/"armstrongUsingRecursion
armstrongUsingRecursion.cpp:8:1: warning: non-void function does not return a value in all control paths [-Wreturn-type]
}
^
1 warning generated.
Enter a number:255
It is not an Armstrong Number%
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 %
```

Time Complexity:

Armstong Number program is O(log n)

Space Complexity:

Armstong Number program is O(log n)

3.Write a program to find the GCD of two numbers using recursive function

```cpp
#include <iostream>
using namespace std;

int hcf(int n1, int n2)
{
    if (n2 != 0)
       return hcf(n2, n1 % n2);
    else
       return n1;
}
int main()
{
   int n1, n2;
   cout << "Enter two positive integers: ";
   cin >> n1 >> n2;
   cout << "H.C.F of " << n1 << " & " <<  n2 << " is: " <<
hcf(n1, n2);
   return 0;
}
```

Output:

Time Complexity:

GCD Program is O(log(min(n1, n2)))

Space Complexity:

GCD program is O(1)

4. Write a program to get the largest element of an array

```cpp
#include <iostream>
using namespace std;
int main() {
  int i, n;
  float arr[100];
  cout << "Enter total number of elements(1 to 100): ";
  cin >> n;
  cout << endl;
  for(i = 0; i < n; ++i) {
    cout << "Enter Number " << i + 1 << " : ";
    cin >> arr[i];
  }
  for(i = 1;i < n; ++i) {
    if(arr[0] < arr[i])
      arr[0] = arr[i];
  }
  cout << endl << "Largest element = " << arr[0];
  return 0;
}
```

Output:

```
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 % cd "/Users/supreethkumarjagarlamudi/Documents/CSA06/" && g++ largestElement.cpp -o largestElem
ent && "/Users/supreethkumarjagarlamudi/Documents/CSA06/"largestElement
Enter total number of elements(1 to 100): 5

Enter Number 1 : 1
Enter Number 2 : 4
Enter Number 3 : 6
Enter Number 4 : 2
Enter Number 5 : 8

Largest element = 8
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 %
```

Time Complexity:

largest Element program is O(n)

Space Complexity:

largest Element program is O(1)

5. Write a program to find the factorial of a number using recursion

```cpp
#include<iostream>
using namespace std;
int factorial(int n) {
  if(n > 1)
    return n * factorial(n - 1);
  else
    return 1;
}
int main() {
  int n;
  cout << "Enter a positive integer: ";
  cin >> n;
  cout << "Factorial of " << n << " = " << factorial(n);
  return 0;
}
```

Output:

Time Complexity:

factorial program is O(n)

Space Complexity:

factorial program is O(n)

6.Write a program for to copy one string to another using recursion

```cpp
#include <iostream>
using namespace std;

void myCopy(char s1[], char s2[], int index = 0)
{
    s2[index] = s1[index];

    if (s1[index] == '\0')
        return;

    myCopy(s1, s2, index + 1);
}

// Driver function
int main()
{
    char s1[100] = "SupreethKumar";
    char s2[100] = "";
    myCopy(s1, s2);
    cout << s2;
    return 0;
}
```

Output:

```
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 % cd "/Users/supreethkumarjagarlamudi/Documents/CSA06/" && g++ copyStringRecursion.cpp -o copySt
ringRecursion && "/Users/supreethkumarjagarlamudi/Documents/CSA06/"copyStringRecursion
SupreethKumar
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 %
```

Time Complexity:

Copy String program is O(n), Where n is the length of the first String

Space Complexity:

Copy String program is O(n), Where n is the length of the first String

7. Write a program to print the reverse of a string using recursion

```cpp
#include <iostream>
using namespace std;

void reverse(char *str, int index, int n)
{
    if(index == n)
    {
        return;
    }
    char temp = str[index];
    reverse(str, index+1, n);
    cout << temp;
}

int main()
{
    char a[] = "SupreethKumar";
    int n = sizeof(a) / sizeof(a[0]);
    reverse(a, 0, n);
    return 0;
}
```

Output:

Time Complexity:

reverse String program is O(n), Where n is the length of the first String

Space Complexity:

reverse String program is O(n), Where n is the length of the first String

8. Write a program to generate all the prime numbers using recursion

```cpp
#include <iostream>
using namespace std;

bool isPrime(int n)
{

    if(n == 1 || n == 0) return false;
    for(int i = 2; i < n; i++)
    {
        if(n % i == 0) return false;
    }
    return true;
}

int main()
{
    int N = 100;

    for(int i = 1; i <= N; i++)
    {
        if(isPrime(i))
        {
            cout << i << " ";
        }
    }

    return 0;
}
```

Output:

```
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 % cd "/Users/supreethkumarjagarlamudi/Documents/CSA06/" && g++ primeNumberRecursionG.cpp -o prim
eNumberRecursionG && "/Users/supreethkumarjagarlamudi/Documents/CSA06/"primeNumberRecursionG
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 %
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 %
```

Time Complexity:

prime numbers generation program is O(n^2)

Space Complexity:

prime numbers generation program is O(1)

9. Write a program to check a number is a prime number or not using recursion

```cpp
#include <iostream>
using namespace std;
bool isPrime(int n, int i = 2)
{
    if (n <= 2) return (n == 2) ? true : false; if (n % i == 0)
return false; if (i * i > n)
       return true;

    return isPrime(n, i + 1);
}

int main()
{
    int n = 34;
    if (isPrime(n))
       cout<< "Prime Number";
    else
       cout<< "Not a Prime";

    return 0;
}
```

Output:

```
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 % cd "/Users/supreethkumarjagarlamudi/Documents/CSA06/" && g++ primeNumbersRecusion.cpp -o primeNumbersRecusion && "/Users/supreethkumarjagarlamudi/Documents/CSA06/"primeNumbersRecusion
Not a Prime
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 %
```

Time Complexity:

prime numbers program is O(sqrt(n))

Space Complexity:

prime numbers program is O(1)

10. Write a program for to check whether a given String is a palindrome or not using recursion

```cpp
#include <iostream>
using namespace std;

bool isPalRec(char str[],
              int s, int e)
{

    if (s == e)
    return true;

    if (str[s] != str[e])
    return false;

    if (s < e + 1)
    return isPalRec(str, s + 1, e - 1);

    return true;
}

bool isPalindrome(char str[])
{
    int n = strlen(str);

    if (n == 0)
        return true;

    return isPalRec(str, 0, n - 1);
}

int main()
{
    char str[] = "geeg";

    if (isPalindrome(str))
    cout << "Yes";
    else
    cout << "No";

    return 0;
}
```

Output:

```
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 % cd "/Users/supreethkumarjagarlamudi/Documents/CSA06/" && g++ palindrome.cpp -o palindrome && "
/Users/supreethkumarjagarlamudi/Documents/CSA06/"palindrome
Yes
supreethkumarjagarlamudi@Supreeths-MacBook-Pro CSA06 %
```