1. Write a program to Print Fibonacci Series using recursion.

Code:

**#include <stdio.h>**

**int fibonacci(int n) {**

**if(n == 0)**

**return 0;**

**else if(n == 1)**

**return 1;**

**else**

**return (fibonacci(n-1) + fibonacci(n-2));**

**}**

**int main() {**

**int n;**

**printf("Enter the number of terms\n");**

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

**printf("Fibonacci Series: ");**

**for (int i = 0; i < n; i++) {**

**printf("%d ", fibonacci(i));**

**}**

**return 0;**

**}**

1. **Write a program to check the given no is Armstrong or not using recursive function.**

**Code:**

**#include<stdio.h>**

**#include<math.h>**

**int check\_ArmstrongNumber(int num)**

**{**

**if(num>0)**

**return (pow(num%10,3) +check\_ArmstrongNumber(num/10));**

**}**

**int main()**

**{**

**int num;**

**printf("Enter a number:");**

**scanf("%d",&num);**

**if(check\_ArmstrongNumber(num)==num)**

**printf("It is an Armstrong Number");**

**else**

**printf("It is not an Armstrong Number");**

**}**

**3.Write a program to find the gcd of 2 numbers using recursion.**

**Code:**

**#include <stdio.h>**

**int hcf(int n1, int n2);**

**int main() {**

**int n1, n2;**

**printf("Enter two positive integers: ");**

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

**printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));**

**return 0;**

**}**

**int hcf(int n1, int n2) {**

**if (n2 != 0)**

**return hcf(n2, n1 % n2);**

**else**

**return n1;**

**}**

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

Code:

#include <stdio.h>

int main() {

int n;

double arr[100];

printf("Enter the number of elements (1 to 100): ");

scanf("%d", &n);

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

printf("Enter number%d: ", i + 1);

scanf("%lf", &arr[i]);

}

for (int i = 1; i < n; ++i) {

if (arr[0] < arr[i]) {

arr[0] = arr[i];

}

}

printf("Largest element = %.2lf", arr[0]);

return 0;

}

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

Code:

#include<stdio.h>

long int multiplyNumbers(int n);

int main() {

int n;

printf("Enter a positive integer: ");

scanf("%d",&n);

printf("Factorial of %d = %ld", n, multiplyNumbers(n));

return 0;

}

long int multiplyNumbers(int n) {

if (n>=1)

return n\*multiplyNumbers(n-1);

else

return 1;

}

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

Code:

#include <bits/stdc++.h>

**using** **namespace** std;

**void** myCopy(**char** s1[], **char** s2[])

{

**int** i = 0;

**for** (i=0; s1[i] != '\0'; i++)

       s2[i] = s1[i];

    s2[i] = '\0';

}

**int** main()

{

**char** s1[100] = "GEEKSFORGEEKS";

**char** s2[100] = "";

    myCopy(s1, s2);

    cout << s2;

**return** 0;

}

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

Code:

# include <stdio.h>

**void** reverse(**char** \*str)

{

**if** (\*str)

   {

       reverse(str+1);

**printf**("%c", \*str);

   }

}

**int** main()

{

**char** a[] = "Geeks for Geeks";

   reverse(a);

**return** 0;

}

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

Code:

**def** isPrime(n, i **=** 2):

**if** (n <**=** 2):

**return** True **if**(n **==** 2) **else** False

**if** (n **%** i **==** 0):

**return** False

**if** (i **\*** i > n):

**return** True

**return** isPrime(n, i **+** 1)

n **=** 15

**if** (isPrime(n)):

    print("Yes")

**else**:

**print**("No")

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

Code: **def** isPrime(n, i **=** 2):

**if** (n <**=** 2):

**return** True **if**(n **==** 2) **else** False

**if** (n **%** i **==** 0):

**return** False

**if** (i **\*** i > n):

**return** True

**return** isPrime(n, i **+** 1)

n **=** 15

**if** (isPrime(n)):

    print("Yes")

**else**:

**print**("No")

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

Code:

**def** isPalindrome(s):

**return** s **==** s[::**-**1]

s **=** "malayalam"

ans **=** isPalindrome(s)

**if** ans:

    print("Yes")

**else**:

    print("No")