**Experiment No :** 15

**Experiment name :** . Write a C program using recursive function calls to evaluate F(x) = x – x3 /3! + x5 /5! – x 7 /7! + ………………….

**Methodology :**

1. factorial(int n): This function calculates the factorial of a given integer 'n' using recursion.
2. f\_series(int x, int n): This function evaluates the series F(x) up to the nth term using recursion. It calculates each term based on the power of x and the sign (alternating between positive and negative), adds them up, and returns the result.
3. The main function reads the value of 'x' and the number of terms 'n' from the user, then calls the f\_series function to evaluate the series F(x) up to the nth term and prints the result.

**Flow-Chart :**

Int i , n;

Float x , sum , t , d;

**Code :**

yes

No

I++

Print : sum , n , x ;

D = (2\*i) \*(2\*i-1);

T = -t\*x\*x/d ;

Sum = sum+t ;

I<n?

scanf("%d", &x);

#include <stdio.h>

// Recursive function to calculate the factorial of a given number

int factorial(int n) {

if (n == 0)

return 1;

else

return n \* factorial(n - 1);

}

// Recursive function to evaluate the series F(x) up to the nth term

double f\_series(int x, int n) {

if (n == 0) {

return 0;

} else {

int denominator = factorial(2 \* n - 1);

double term = (n % 2 == 0) ? -1 : 1;

term \*= (double)(x \* x \* x) / denominator;

return term + f\_series(x, n - 1);

}

}

int main() {

int x, n;

printf("Enter the value of x: ");

scanf("%d", &x);

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

scanf("%d", &n);

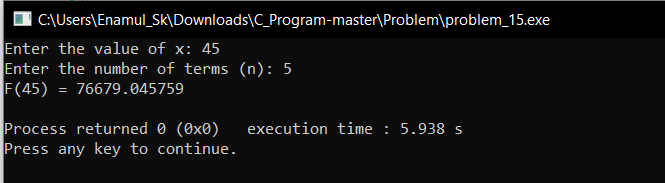
double result = f\_series(x, n);

printf("F(%d) = %lf\n", x, result);

return 0;

}

**Output:**



**Result discussion :**

The messages that we have used in this programming is that inside the Recursive function it will return value if it is and if it is not then n\*factorial(n-1) What we have done and below we are again using the same message where we have done it as it is Figured things out.