

Assignment – 8

The objective of this assignment is to learn how to use user define function in the program
Assignment:

1. Write a C program to find the factorial of a number and also find the value of nC_r using this function.

Factorial of a number using function:

STEP 1: **Declare a function** 'fact' with input parameter (n) before main function

STEP 2: In main, take the number(n) as user input

STEP 3: **Call the function** and pass the input parameter (n)

[factorial = fact (n)]

STEP 4: In **function definition** initialize a variable 'f' with 1

STEP 5: Start a for loop from i=1 to the number(n)

STEP 6: Multiply the value of 'f' with 'i' in the loop

STEP 7: Return the final value of 'f' to the main function

STEP 8: Print the value of 'factorial' in the main function

Find the value of nC_r using 'factorial function':

[**Hint:** Take another user input r. Call the 'fact' function using the following equation:

$$ncr = \text{fact}(n) / (\text{fact}(r) * \text{fact}(n-r))$$
]

2. Write a C Program to find the sum of two matrices using function.

STEP 1: **Declare a function** 'matrixadd' with input parameters,

[void matrixadd (int [][][10], int[][10], int[][10], int, int);]

STEP 2: In main, define 3 matrices : matrix1, matrix2, matrix3 and the required variables

[int matrix1[10][10], matrix2[10][10], matrix3[10][10], row, col, i, j;]

STEP 3: Take number of rows and columns of the matrices as input

STEP 4: Take 2 user defined matrix inputs (matrix1, matrix2) in 2D array.

```
printf("\nEnter the elements of the 1st matrix: ");
for(i=0;i<row;i++)
{
    for(j=0;j<col;j++)
    {
        printf("\nmatrix1[%d][%d]=",i,j);
        scanf("%d",&matrix1[i][j]);
    }
}
```

[Similarly take the elements of the 2nd matrix.]

STEP 5: **Call the function** and pass the input parameters (matrix1, matrix2, matrix3)
[matrixadd(matrix1,matrix2,matrix3,row,col);]

STEP 6: In **function definition** add the 2D arrays: matrix1 and matrix2 using 2 nested for loops i and j.

```
void matrixadd(int matrix1[][10],int matrix2[][10],int matrix3[][10],int m, int n)
{
    int i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            matrix3[i][j]=matrix1[i][j]+matrix2[i][j];
        }
    }
}
```

STEP 7: Print the 2D array matrix3 as output in the main function.

```
printf("\nThe sum of the two matrices:\n");
for(i=0;i<row;i++)
{
    for(j=0;j<col;j++)
    {
        printf("%3d",matrix3[i][j]);
    }
    printf("\n");
}
```

3. **Write a C program to find power of any number using recursive function.**

STEP 1: **Declare a recursive function** 'power' with input parameters (base and exponent)

STEP 2: Take 'base' and 'exponent' as user inputs in main.

STEP 3: **Call the recursive function** and pass the input parameters

[result = power (base,exponent)]

STEP 4: In **recursive function definition** check, If the value of exponent is 0 then
return 1 to main

STEP 5: If the value of exponent is greater than 0

return (base * power(base, exponent-1)) [calling the function 'power'
recursively]

STEP 6: If the value of exponent is less than 0,

return (1 / pow(base, -exponent)) [calling the function 'power'
Recursively]

STEP 7: Print the value of 'result' in main().

4. Write a function in C program to sort all elements of an array in ascending order using bubble sort technique.

[Bubble sort is a comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.]

STEP 1: **Declare a function 'bubble'** with input parameters - array and size of the array
[void bubble (int [], int)]

STEP 2: Take the array size and array elements as inputs in the main function.

STEP 3: **Call the 'bubble' function** from main() and pass the input parameters (array name and size of the array)

STEP 4: In **function definition** run an outer for loop (say i) from 0 to less than (size-1)

STEP 5: Run an inner for loop (say j) from 0 to less than (size-1-i)

STEP 6: Inside the inner for loop, check **if jth element of the array is greater than (j+1)th element of the array, swap the elements.**

STEP 7: Print the array after completion of sorting

5. Write a C program to sort all elements of an array in ascending order using merge sort technique.

[Merge sort first divides the array into equal halves and then combines them in a sorted manner.]

STEP 1:

6. Write a C program to find the GCD of two numbers using a recursive function, and also find the GCD of three numbers using this function.

7. Write a C program to find the Fibonacci series up to n term using recursive function

Practice:

8. Write a C program to find maximum and minimum elements in an array using a recursive function.

9. Write a C program to sort all elements of an array in ascending order using quick sort technique.

10. Write a C Program to count the frequency of array elements in a 1-D array

11. Write a C program to solve Tower of Hanoi problem

[Objective: Move n disks from 1 rod to another rod with the help of an intermediate rod.

[NOTE: Implement the previous assignments using function call]