

Course Name: O Level (2nd Sem B1 Batch)**Subject:** C Language**Topic:** Recursion in C Continued..**Date:** 22-April-2020**Recursion**

Recursion is a process by which a function calls *itself* directly or indirectly. The corresponding function is called as **recursive function**. Using recursive algorithms, certain complex problems can be solved quite easily.

Example 1: Write a recursive function to print the table of any input number.

```
#include<stdio.h>
#include<conio.h>
void tab(int,int);
void main()
{
    int a,b=10;
    clrscr();
    printf("enter any no.");
    scanf("%d",&a);
    tab(a,b);
    getch();
}
void tab (int x, int y)
{
    int i;
    if(y==0)
        return;
    else
        i=x*y;
    tab(x,--y);
    printf("\n%d",i);
}
```

Example 2: Write a recursive function to print the sum of any digit number.

```
#include<stdio.h>
#include<conio.h>
int sum(int);
void main()
{
    int x,y;
    clrscr();
    printf("enter any digit no.");
    scanf("%d",&x);
    y=sum(x);
    printf("Addition of No.= %d",y);
    getch();
}
sum (int x1)
{
    static int i;
    if(x1==0)
        return(i);
    i=i+(x1%10);
    x1=sum(x1/10);
}
```

```
}
```

Example 3: Write a recursive function to convert any decimal number to its binary equivalent.

```
#include <stdio.h>
#include <conio.h>
void change (int, int);
main()
{
    int n, base;
    clrscr();
    printf("enter the number and base");
    scanf("%d%d", &n, &base);
    change(n, base);
    getch();
}
void change(int n1, int b)
{
    int x;
    if(n1==0)
        return;
    x=n1%b;
    change(n1/b, b);
    printf("%d", x);
}
```

Example 4: Write a program in C to count the digits of a given number using recursion.

```
#include <stdio.h>
#include <conio.h>
int nod(int);
void main()
{
    int n1, ctr;
    printf(" Input a number : ");
    scanf("%d", &n1);
    ctr = nod(n1);
    printf("\nThe number of digits in the number is : %d", ctr);
    getch();
}
int nod(int n1)
{
    static int ctr=0;
    if(n1!=0)
    {
        ctr++;
        nod(n1/10);
    }
    return ctr;
}
```

Example 5: Write a program in C to find GCD of two numbers using recursion.

```
#include <stdio.h>
#include <conio.h>
int findGCD(int, int);
void main()
{
    int num1, num2, gcd;
```

```

printf("\nInput 1st number: ");
scanf("%d",&num1);
printf("\nInput 2nd number: ");
scanf("%d",&num2);
gcd = findGCD(num1,num2);
printf("\n The GCD of %d and %d is: %d\n\n",num1,num2,gcd);
getch();
}
int findGCD(int a,int b)
{
    while(a!=b)
    {
        if(a>b)
            return findGCD(a-b,b);
        else
            return findGCD(a,b-a);
    }
    return a;
}

```

Example 6: Write a program in C to find the LCM of two numbers using recursion.

```

#include <stdio.h>
#include <conio.h>
int lcm (int, int);
void main()
{
    int n1, n2, l;
    printf("\nInput 1st number for LCM : ");
    scanf("%d", &n1);
    printf("\nInput 2nd number for LCM : ");
    scanf("%d", &n2);
    if(n1 > n2)
        l = lcm(n2, n1)
    else
        l = lcm(n1, n2);
    printf("\nThe LCM of %d and %d : %d\n\n", n1, n2, l);
    getch();
}
int lcm (int a, int b)
{
    static int m = 0;
    m += b;
    if((m % a == 0) && (m % b == 0))
    {
        return m;
    }
    else
    {
        lcm(a, b);
    }
}

```

Try Yourself:

1. Write a recursive function to print the multiple of any digit number.
2. Write a recursive function to check any digit number is palindrome or not.
3. Write a recursive function to convert any decimal number to its octal equivalent.