

- **Name** : Kedarnath Prasad Chavan
- **PRN** : 21610047
- **Batch** : S-2
- **Course** : Data Structures
- **Branch** : Information Technology

Recursion Programs

1. Fibonacci Series

```
#include <stdio.h>
int fabonacci (int n);
int main ()
{
    int n,i;
    printf ("\nEnter the number : ");
    scanf ("%d",&n);
    printf ("\nThe fabonacci numbers are : ");
    for (i=0; i<n; i++)
    {
        printf ("%d ",fabonacci(i));
    }
    printf ("\n");
    return 0;
}

int fabonacci (int n)
{
    if (n==0)
        return 0;
    if (n==1)
        return 1;
    return (fabonacci(n-1) + fabonacci(n-2));
}
```

```
kedarnath@TUX:~/programs/c-programs$ gcc fabonacci-rec.c
kedarnath@TUX:~/programs/c-programs$ ./a.out

Enter the number : 10

The fabonacci numbers are : 0 1 1 2 3 5 8 13 21 34

kedarnath@TUX:~/programs/c-programs$ |
```

2. Factorial of a number

```
#include <stdio.h>
int factorial (int);
int factorial (int n)
{
    if (n==1)
        return 1;
    else
        return (n * factorial(n-1));
}

int main ()
{
    int n,fact;
    printf ("\nEnter the value : ");
    scanf ("%d",&n);
    fact = factorial (n);
    printf ("\nThe factorial value is : %d\n\n",fact);
    return 0;
}
```

```
kedarnath@TUX:~/programs/c-programs$ gcc fact.c
kedarnath@TUX:~/programs/c-programs$ ./a.out

Enter the value : 10

The factorial value is : 3628800

kedarnath@TUX:~/programs/c-programs$ |
```

3. GCD of two numbers

```
#include <stdio.h>
int gcd (int a, int b);

int main ()
{
    int a,b,g;
    printf ("\nEnter the numbers : ");
    scanf ("%d %d",&a, &b);
    g = gcd(a,b);
    printf ("\nThe GCD of these numbers is : %d\n\n",g);
    return 0;
}

int gcd (int a, int b)
{
    if (a==0)
        return b;
    if (b==0)
        return a;
    return gcd (b,a%b);
}
```

```
kedarnath@TUX:~/programs/c-programs$ gcc GCD-rec.c
kedarnath@TUX:~/programs/c-programs$ ./a.out

Enter the numbers : 100
250

The GCD of these numbers is : 50
kedarnath@TUX:~/programs/c-programs$ |
```

4. Ackerman function

```
#include <stdio.h>
int ackerman (int a, int b);

int main ()
{
    int x,y,a;
    printf ("\nEnter two numbers : ");
    scanf ("%d %d",&x, &y);
    a = ackerman (x,y);
    printf ("\nThe Ackerman number is : %d\n\n",a);
    return 0;
}

int ackerman (int a, int b)
{
    if (a==0)
        return (b+1);
    else if (b==0)
        return ackerman(a-1,1);
    else
        return ackerman(a-1,ackerman(a,b-1));
}
```

```
kedarnath@TUX:~/programs/c-programs$ gcc ackerman-function.c
kedarnath@TUX:~/programs/c-programs$ ./a.out

Enter two numbers : 2
6

The Ackerman number is : 15

kedarnath@TUX:~/programs/c-programs$ |
```

5. Tower of Hanoi

```
#include <stdio.h>
void towers(int, char, char, char);
int main()
{
    int num;
    printf("Enter the number of disks : ");
    scanf("%d", &num);
    printf("The number of moves needed in the Tower of Hanoi are : \n");
    towers(num, 'A', 'C', 'B');
    return 0;
}
void towers(int num, char from, char to, char auxiliary)
{
    if (num == 1)
    {
        printf("\n Moving disk 1 from pole %c to pole %c\n", from, to);
        return;
    }
    towers(num - 1, from, auxiliary, to);
    printf("\n Moving disk %d from pole %c to pole %c\n", num, from, to);
    towers(num - 1, auxiliary, to, from);
}
```

```
kedarnath@TUX:~/programs/c-programs$ gcc tower-of-hanoi.c
kedarnath@TUX:~/programs/c-programs$ ./a.out
Enter the number of disks : 3
The number of moves needed in the Tower of Hanoi are :

Moving disk 1 from pole A to pole C
Moving disk 2 from pole A to pole B
Moving disk 1 from pole C to pole B
Moving disk 3 from pole A to pole C
Moving disk 1 from pole B to pole A
Moving disk 2 from pole B to pole C
Moving disk 1 from pole A to pole C
```