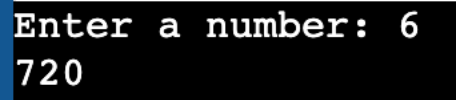


AIM:	Apply the concept of recursion to solve a given problem
Program 1	
PROBLEM STATEMENT :	Write a recursive function to find the factorial of a number and test it.
PROGRAM:	<pre> #include <stdio.h> //Declared the Function int factorial(int num); int main() { printf("Enter a number: "); int num; scanf("%d",&num); printf("%d",factorial(num)); return 0; } //This function calculate the factorial of number entered by user int factorial(int num){ if (num<=1) return 1; else return num*factorial(num-1); } </pre>
RESULT: 	
Program 2	
PROBLEM STATEMENT :	Write a recursive function which returns the nth term of the Fibonacci series. Call it from main() to find the 1st n numbers of the Fibonacci series.
PROGRAM:	<pre> #include <stdio.h> //Declared the function int fibonacci(int num); int main() { int num; printf("Enter the number: "); scanf("%d",&num); for (int i=1;i<=num;i++){ printf("%d, ",fibonacci(i)); } } </pre>

	<pre> } return 0; } //function which calculate the nth term of fibonacci series int fibonacci(int num){ if (num==1) return 0; else if (num==2) return 1; else return fibonacci(num-1)+fibonacci(num-2); } </pre>
--	--

RESULT:

```

Enter the number: 6
0, 1, 1, 2, 3, 5,

```

Program 3

PROBLEM STATEMENT:	<p>Given a number n, print following a pattern without using any loop.</p> <p>Example:</p> <p>Input: n = 16</p> <p>Output: 16, 11, 6, 1, -4, 1, 6, 11, 16</p> <p>Input: n = 10</p> <p>Output: 10, 5, 0, 5, 10</p>
PROGRAM:	<pre> #include <stdio.h> //Declared Fuction int print_num_reverse(int); int print_num_for(int,int); int main() { printf("Enter a number: "); int num; scanf("%d",&num); int last_num=print_num_reverse(num);//storing the negative value or 0 to last_num print_num_for(last_num,num); return 0; } //This function will print till negavtive fuction int print_num_reverse(int num){ if (num<=0) return num; printf("%d, ",num); return print_num_reverse(num-5); } </pre>

	<pre> } //This function will print from negative number to entered number int print_num_for(int num, int max){ printf("%d, ",num); if (num==max) return 1; return print_num_for(num+5,max); } </pre>
--	--

RESULT:

Case 1:

```

Enter a number: 16
16, 11, 6, 1, -4, 1, 6, 11, 16,

```

Case 2:

```

Enter a number: 10
10, 5, 0, 5, 10,

```

Program 4

PROBLEM STATEMENT:	<p>Ackerman's function is defined by:</p> $A(m,n) = n+1 \text{ if } m=0$ $= A(m-1, 1) \text{ if } m \neq 0 \text{ and } n=0$ $= A(m-1, A(m,n-1)) \text{ if } m \neq 0 \text{ and } n \neq 0$ <p>Write a function which given m and n returns A(m,n). Tabulate the values of A(m,n) for all m in the range 1 to 3 and all n in the range 1 to 6.</p>
PROGRAM:	<pre> #include <stdio.h> //Declared the Function int Ackerman(int,int); int main() { //Print all the ackerman function for (int i=1;i<=3;i++){ for (int j=1;j<=6;j++){ printf("Ackerman(%d,%d) is %d\n",i,j,Ackerman(i,j)); } } return 0; } int Ackerman(int m,int n){ if (m==0) return n+1; else if (m!=0 && n==0) return Ackerman(m-1,1); } </pre>

	<pre> else return Ackerman(m-1,Ackerman(m,n-1)); } </pre>
--	---

RESULT:

```

Ackerman(1,1) is 3
Ackerman(1,2) is 4
Ackerman(1,3) is 5
Ackerman(1,4) is 6
Ackerman(1,5) is 7
Ackerman(1,6) is 8
Ackerman(2,1) is 5
Ackerman(2,2) is 7
Ackerman(2,3) is 9
Ackerman(2,4) is 11
Ackerman(2,5) is 13
Ackerman(2,6) is 15
Ackerman(3,1) is 13
Ackerman(3,2) is 29
Ackerman(3,3) is 61
Ackerman(3,4) is 125
Ackerman(3,5) is 253
Ackerman(3,6) is 509

```

Program 5

PROBLEM STATEMENT:	Write a program to find the value of y using recursive function, where $y=x^n$
PROGRAM:	<pre> #include <stdio.h> //Declared The function int power_fuc(int num,int power); int main() { printf("Enter the number and power: "); int num,power; scanf("%d %d",&num,&power); int result=power_fuc(num,power); printf("%d raise to power %d is: %d",num,power,result); return 0; } //Fuction which Calculate the power of a number int power_fuc(int num,int power){ if (power==0) return 1; return num*power_fuc(num,power-1); } </pre>

RESULT:

```
Enter the number and power: 16 2
16 raise to power 2 is: 256
```

CONCLUSION:

We learned to apply the concept of recursion to solve a given problem