

/\*

1. Recursive functions or recursion is a process when a function calls a copy of itself to work on a smaller problem.
2. Any function which calls itself is a recursive function.
3. This makes the life of programmer easy by dividing a given problem into easier problems.
4. A termination condition is imposed on such functions to stop them executing copies of themselves forever.
5. Any problem that can be solved recursively can also be solved iteratively.

Why recursions?

1. Some problems are best suited to be solved by recursions.

Example: Tower of hanoi, Fibonacci series, Factorial etc.

Factorial calculation:

1. The case at which the function doesn't occur is called base case.
2. The instances where the function keeps calling itself to perform subtask, is called

*recursive case.*

*3. For factorial calculation, the base case occurs at the parameter value of 0 and 1.*

```
*/  
  
#include <stdio.h>  
  
int factorial(int number){  
    if(number==0||number==1){  
        return 1;  
    }  
    else{  
        return number*factorial(number-1);  
    }  
}  
  
int main()  
{  
    int num;  
    printf("Enter a number:");  
    scanf("%d",&num);  
    printf("Factorial of %d is %d.\n",num,factorial(num));  
    return 0;  
}
```

}