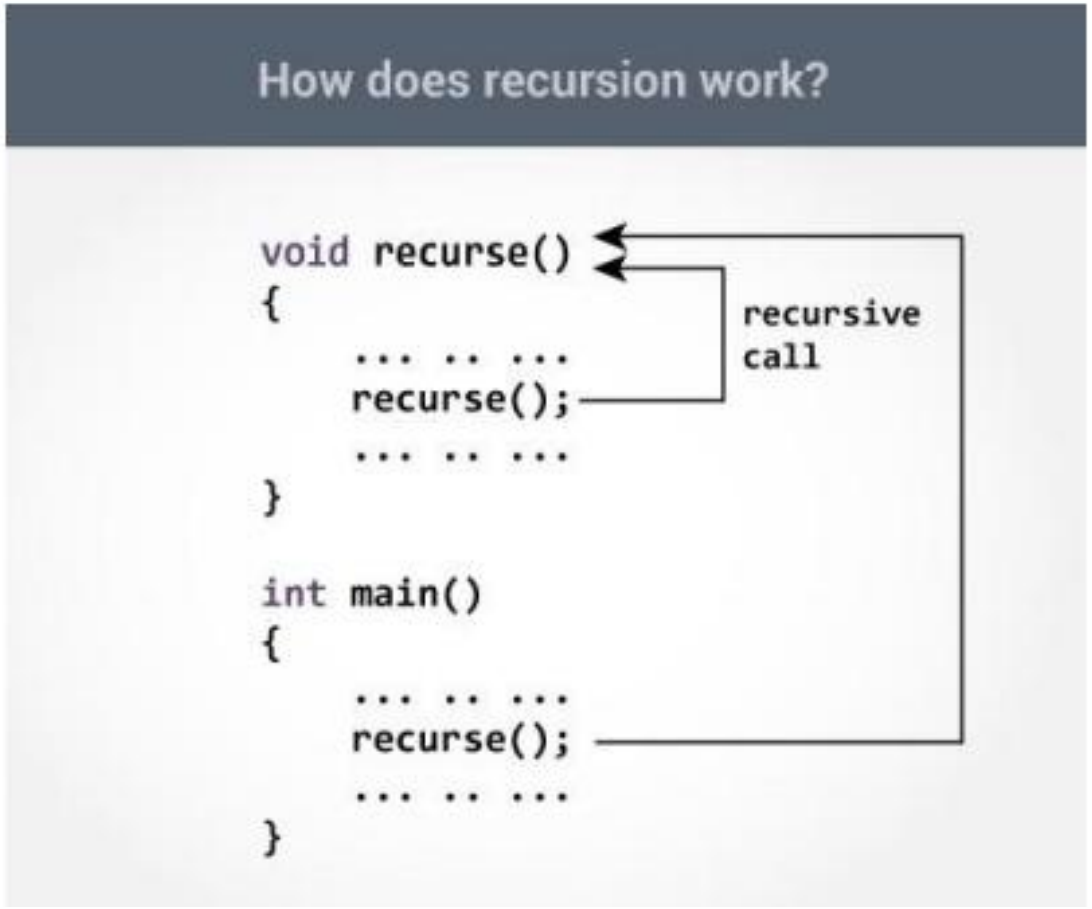
Recursion

Dr. Asif Uddin Khan

C Recursion

- A function that calls itself is known as a recursive function. And, this technique is known as recursion.
- The recursion continues until some condition is met to prevent it.



C Recursion

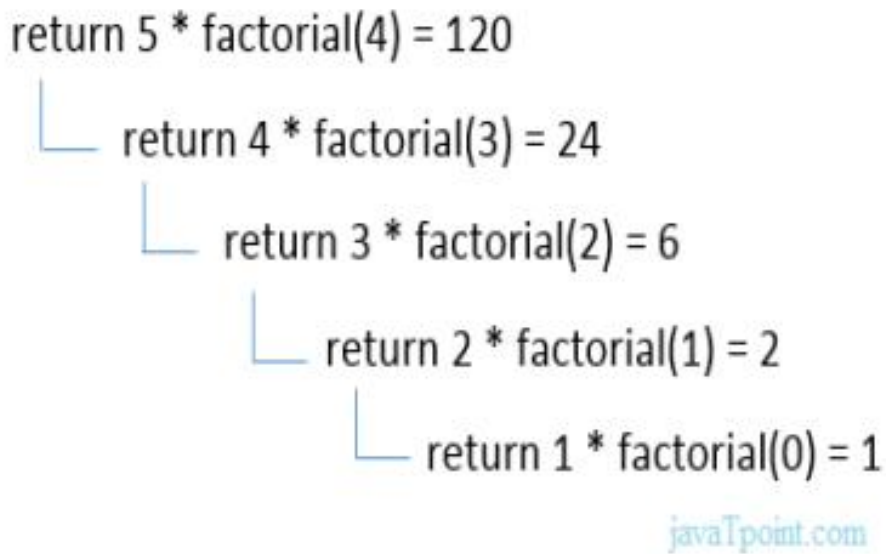
- To prevent infinite recursion, if...else statement (or similar approach) can be used where one branch makes the recursive call, and other doesn't.
- A recursive function calls itself with smaller size argument.
- There should be some terminating condition
- The recursion continues until the condition is met.
- To prevent infinite recursion, if...else statement (or similar approach) can be used where one branch makes the recursive call, and other doesn't.

factorial using recursion

```
//factorial using recursion
#include <stdio.h>
int fact (int);
int main()
{
   int n,f;
   printf("Enter the number");
   scanf("%d",&n);
   f = fact(n);
   printf("factorial = %d",f);
}
```

```
int fact(int n)
{
   if (n==0)
   {
      return 1;
   }
   else if ( n == 1)
   {
      return 1;
   }
   else
   {
      return n*fact(n-1);
   }
}
```

factorial using recursion



1 * 2 * 3 * 4 * 5 = 120

Fig: Recursion

Example: Sum of Natural Numbers Using Recursion

```
//sum of natural numbers using
    recursion
#include <stdio.h>
int sum(int n);
int main() {
   int number, result;
   printf("Enter a positive integer: ");
   scanf("%d", &number);
   result = sum(number);
   printf("sum = %d", result);
   return 0;
}
int sum(int n) {
   if (n != 0)
      // sum() function calls itself
      return n + sum(n-1);
   else
      return n;
}
```