EDUCBA
(https://www.educba.com/software-development/)

← (https://www.educba.com/hash-table-in-c/)

→ (https://www.educba.com/power-function-in-c/)

Introduction to Recursive Function in C

The process of repeating the items in a similar way as it was before is known as recursio function is said to be recursive if it is called within itself. Recursion is supported by (https://www.educba.com/recursion-in-c-plus-plus/) the programming language C. Below

two conditions that are critical for implementing recursion in C:

In this way, we can implement a recursive function in the C programming language (https://www.educba.com/career-in-c-programming/). These functions are useful for solving money mathematical problems that require a similar process to be called several times. Examples of such problems are calculating the factorial of a number of Fibonacci series (https://www.educba.com/fibonacci-series-in-javascript/) generation.

Syntax:

```
int  fun(a1)
{
If(base_condition) return val;
fun(a2);
}
```

How Recursive Function Works in C?

Recursive functions are the way to implement the equation in C programming language. A recursive function is called with an argument passed into it say n, memory in the stack is allocated to the local variables as well as the functions. All the operations present in the function are performed using that memory. The condition for exit is checked if it fulfills.

compiler detects a call to another function it immediately allocates new memory on the top of memory allocated to that function gets cleared. similarly, the new value gets calculated in the calling function and IT returns to the super calling function. This way recursive calls are made to the function delete reaches the first function and the whole stack memory gets cleared and output is returned. Incase base condition or exit condition is not specified in the function then recursive calls to the function can lead to an infinite loop.

Example of Recursive Function

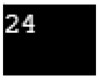
Now we will be going to see the examples of Recursive Function in C

Code:

```
#include <stdio.h>
int  fun(int n)
{
if(n==1) return 1 ; //exit or base condition which gives an idea when to exit this loop.
return n*fun(n-1); //function is called with n-1 as  it's argument .
//The value returned is multiplied with the argument passed in calling function.
}
int main(){
int test=4;
int result =0;


result =fun(test);
```

Output:



```
24
```

Explanation of Above Code

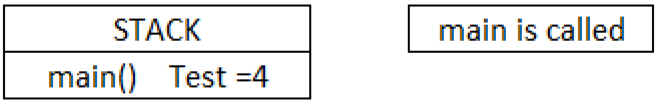
The above-given example is of finding the factorial of a number and from the first call 4*3*2*1 is returned. Thus result in main function stores 24 and prints that on output.

Memory Allocation of Recursive Function

Each call to a function in c language results in memory allocation on the top of a stack. When a recursive function is called memory is allocated to it on the top of the memory that has been allocated to the calling function with all the different copy of local variables are created for each call to the function.
What is the base condition is reached, the memory allocated to the function gets destroyed and pointer returns to the calling function? this process is repeated then the first calling function and at last, the stack memory gets empty.

In the above-given example to calculate the factorial of a number below is the scenario for memory allocation.

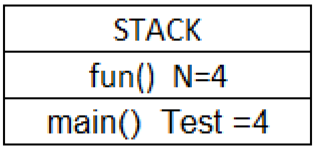
Step – 1



STACK
main()   Test =4

main is called

Step – 2

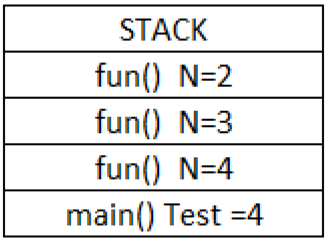


STACK
fun() N=4
main() Test =4

Step – 3

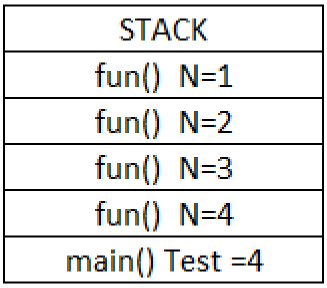
STACK

EDUCBA
(https://www.educba.com/software-development/)



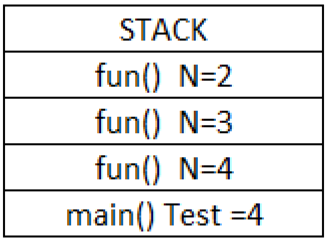
STACK
fun() N=2
fun() N=3
fun() N=4
main() Test =4

Step – 5



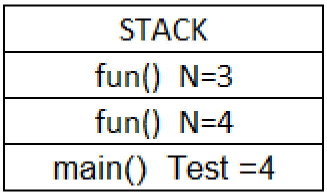
STACK
fun() N=1
fun() N=2
fun() N=3
fun() N=4
main() Test =4

Step – 6



STACK
fun() N=2
fun() N=3
fun() N=4
main() Test =4

Step – 7

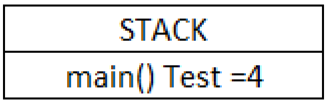


STACK
fun() N=3
fun() N=4
main() Test =4

Step – 8

(https://www.educba.com/software-development/)

Step – 9



STACK
main() Test =4

Types of Recursion

There are two types of recursion in C programming that are given below:

1. Tail and Non-Tailed Recursion

The above-given type of recursion is explained below:

- **Tail Recursion**

It is a type of recursive function recursion call in the function that is the last action to be done in the definition of the function. Means recursive call occurs after everything else logic in the function gets implemented.

Using a tail recursion in our program in hansis the performance of the program and also reduces the memory usage of so function. It is so because as other logic in the function has been implemented to the memory allocated to the calling function can be removed from the stack and reused.

Code:

```
int fun1( ) {
```

EDUCBA (https://www.educba.com/software-development/)

```
{
fun1(4);
}
```

- **Non-Tailed Recursion**

This type of recursion recursive collage made in the middle of the function definition. Men's pants recursion is completed and the values returned to the calling function there are more steps to be performed thus the memory cannot be cleared.

Code:

```
int fun1(n){
printf("the result is ");
return n* fun1(n-1);
}
void main(){
fun1(4);
}
```

2. Direct and Indirect Recursion

The above-given type of recursion is explained below:

- **Indirect Recursion**

Indirect recursion is said to occur when a particular function is called in recursive way medium

```
int fun1(){
fun2();
}
int fun2(){
fun1(); // calling the procedure recursively using another
function.
}
void main(){
fun1();
}
```

- **Direct Recursion**

Direct recursion is said to occur when the recursive call to the function is made within its own definition.'

Code:

```
int fun1(){
fun1();
}
void main(){
fun1();
}
```

QUIZ

Conclusion

(https://www.educba.com/software-development/)

It is important to mention a base condition for the recursive function (https://www.educba.com/recursive-function-in-javascript/). Memory and time requirements are greater for the recursive program as compared to the iterative ones, thus must be used carefully.

Recommended Articles

This is a guide to Recursive Function in C. Here we discuss the basic concept, working, types, memory allocation and examples of Recursive Function in C. You may also look at the following articles to learn more –

1. Arrays in C Programming (https://www.educba.com/arrays-in-c-programming/)
2. Palindrome in C Program (https://www.educba.com/palindrome-in-c-program/)
3. Patterns in C Programming (https://www.educba.com/patterns-in-c-programming/)
4. Difference Between C vs C++ (https://www.educba.com/c-vs-c-plus-plus/)

EDUCBA (https://www.educba.com/software-development/)

About Us

Blog (https://www.educba.com/blog/?source=footer)

Who is EDUCBA? (https://www.educba.com/about-us/?source=footer)

Sign Up (https://www.educba.com/software-development/signup/?source=footer)

Corporate Training (https://www.educba.com/corporate/?source=footer)

Certificate from Top Institutions (https://www.educba.com/educbalive/?source=footer)

Contact Us (https://www.educba.com/contact-us/?source=footer)

Verifiable Certificate (https://www.educba.com/software-development/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/software-development/reviews/?source=footer)

Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

QUIZ

Apps

EDUCBA (https://www.educba.com/software-development/)

Free Courses (https://www.educba.com/software-development/free-courses/?source=footer)

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

Python Tutorials (https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer)

All Tutorials (https://www.educba.com/software-development/software-development-tutorials/?source=footer)

Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

Become a Selenium Automation Tester (https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer)

Become an IoT Developer (https://www.educba.com/software-development/courses/iot-course/?source=footer)

ASP.NET Course (https://www.educba.com/software-development/courses/asp-net-course/?source=footer)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/ph course/?source=footer)

EDUCBA
(https://www.educba.com/software-development/)

QUIZ