**Direct Recursion**[#](https://www.educative.io/courses/recursion-for-coding-interviews-in-cpp/BnKojpzLl2W#Direct-Recursion)

If a function calls itself, it’s known as **direct recursion**. This results in a one-step recursive call: the function makes a recursive call inside its own function body.

**int** num()

{

...

...

**int** num();

## Indirect Recursion[#](https://www.educative.io/courses/recursion-for-coding-interviews-in-cpp/BnKojpzLl2W#Indirect-Recursion)

If the function f1 calls another function f2 and f2 calls f1 then it is **indirect recursion** (or mutual recursion).

This is a two-step recursive call: the function calls another function to make a recursive call.

|  |
| --- |
| **Note:** For indirect recursion, both the functions need to be declared **before** they are defined. |

**int** num()

{

...

...

**int** sum();

}

**int** sum()

{

...

...

**int** num();

}

If method A calls method B, method B calls method C, and method C calls method A we call the methods A, B and C **indirectly recursive** or **mutually recursive**.

Indirect recursion occurs when a method invokes another method, eventually resulting in the original method being invoked again.

Chains of calls in indirect recursion can contain multiple methods, as well as branches, i.e. in the presence of one condition one method to be called, and provided a different condition another to be called.

The depth of indirection may vary.

Indirect recursion requires the same attention to base cases.

|  |  |
| --- | --- |
| Direct Recursion | Indirect Recursion |
| In the direct recursion, only one function is called by itself. | In indirect recursion more than one function are by the other function and number of times. |
| direct recursion makes overhead. | The indirect recursion does not make any overhead as direct recursion |
| The direct recursion called by the same function | While the indirect function called by the other function |
| In direct function, when function called next time, value of local variable will stored | but in indirect recursion, value will automatically lost when any other function is called local variable |
| Direct function engaged memory location | while local variable of indirect function not engaged it |
| Structure of direct function  **int** num()  {  ...  ...  **int** num();  } | Structure of indirect function  **int** num()  {  ...  ...  **int** sum();  }  **int** sum()  {  ...  ...  **int** num();  } |