Difference Between Recursion and Iteration

www.differencebetween.com

Key Difference - Recursion vs Iteration

Recursion and Iteration can be used to solve programming problems. The approach to solving the problem using recursion or iteration depends on the way to solve the problem. The **key difference** between recursion and iteration is that **recursion is a mechanism to call a function within the same function while iteration is to execute a set of instructions repeatedly until the given condition is true.** Recursion and Iteration are major techniques for developing algorithms and building software applications.

What is Recursion?

When a function calls itself within the function, it is known as Recursion. There are two types of recursion. They are finite recursion and infinite recursion. Finite recursion has a terminating condition. Infinite recursion does not have a terminating condition.

Recursion can be explained using the program to calculate factorials.

n! = n * (n-1)!, if n>0

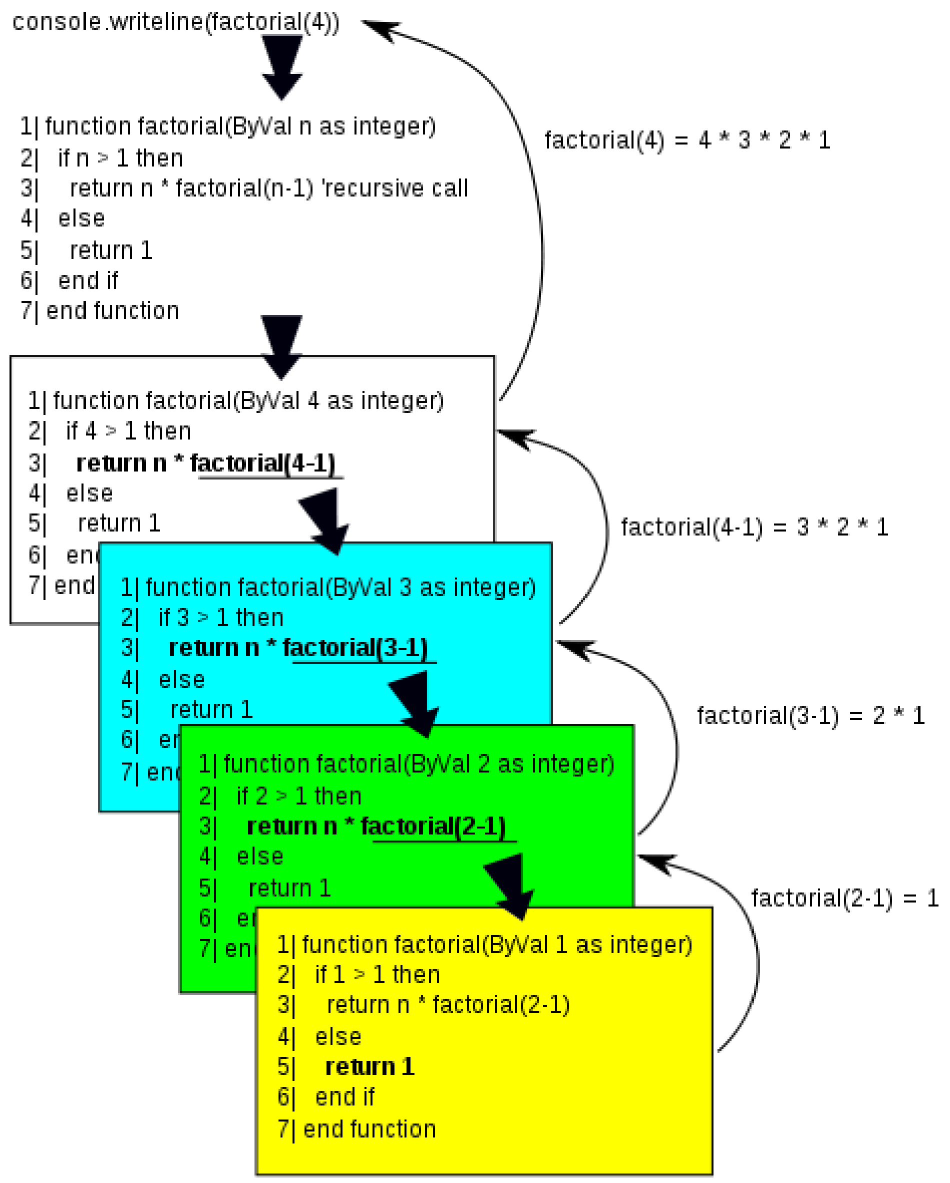
n! =1 ,if n=0;

Refer the bellow code to calculate factorial of 3(3!= 3*2*1).

```
intmain () {

int value =factorial (3);

printf("Factorial is %d\n", value);

return 0;

}

intfactorial (intn) {

if(n==0) {

return 1;
}

else {

return n* factorial(n-1);

}

}
```

When calling factorial (3), that function will call factorial (2). When calling factorial (2), that function will call factorial (1). Then factorial (1) will call factorial (0). factorial (0) will return 1. In the above program, n==0 condition in the "if block" is the base condition. According to the Likewise, the factorial function is called again and again.

Recursive functions are related with the stack. In C, the main program can have many functions. So, main () is the calling function, and the function which is called by the main program is the called function. When the function is called, the control is given to the called function. After the function execution is completed, the control is returned to main.Then the main program continues. So, it creates an activation record or a stack frame to continue execution.



```
console.writeline(factorial(4))
```

```
function factorial(ByVal n as integer)
  if n > 1 then
    return n * factorial(n-1) 'recursive call
  else
    return 1
  end if
end function
```

```
function factorial(ByVal 4 as integer)
  if 4 > 1 then
    return n * factorial(4-1)
  else
    return 1
  en
end
```

```
function factorial(ByVal 3 as integer)
  if 3 > 1 then
    return n * factorial(3-1)
  else
    return 1
  er
end
```

```
function factorial(ByVal 2 as integer)
  if 2 > 1 then
    return n * factorial(2-1)
  else
    return 1
  e
en
```

```
function factorial(ByVal 1 as integer)
  if 1 > 1 then
    return n * factorial(2-1)
  else
    return 1
  end if
end function
```

Figure 01: Recursion

In the above program, when calling factorial (3) from main, it creates an activation record in the call stack. Then, factorial (2) stack frame is created on top of the stack and so on. The activation record keeps information about local variables etc. Each time the function is called, a new set of local variables are created on the top of the stack. These stack frames can slow down the speed up. Likewise in recursion, a function calls itself.The number of times finds the time complexity for a recursive function, the function is called. The time complexity of one function call is O(1). For n number of recursive calls, the time complexity is O(n).

What is Iteration?

Iteration is a block of instructions which repeats again and again till the given condition is true. Iteration can be achieved using “for loop”, “do-while loop” or “while loop”. “for loop” syntax is as follows.

```
for (initialization; condition; modify) {

// statements;

}
```

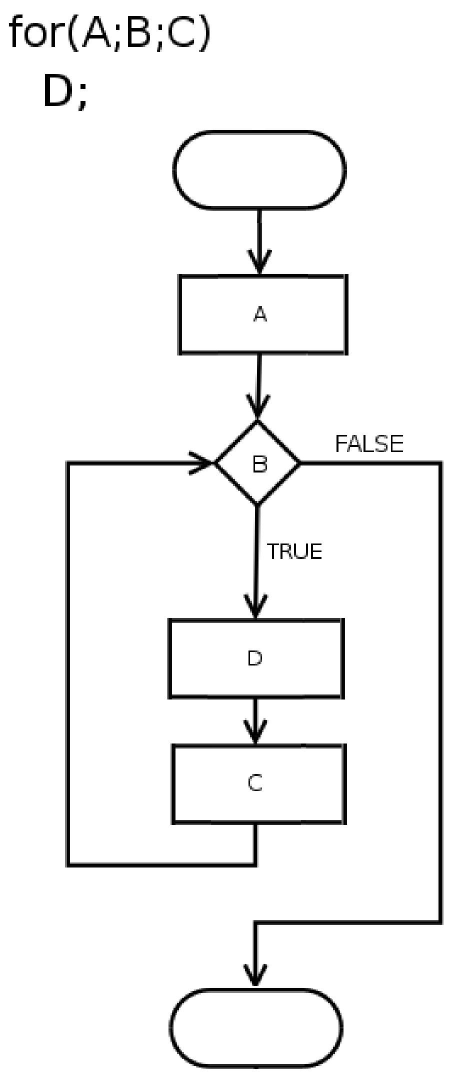


Figure 02: “for loop flow diagram”

Initialization step executes first. This step is to declare and initialize loop control variables.If the condition is true, the statements inside the curly braces execute. Those statements execute till the condition is true. If the condition is false, the control goes to the next statement after the “for loop”. After executing the statements inside the loop, the control goes to modify section. It is to update the loop control variable. Then the condition is checked again. If the condition is true, the statements inside the curly braces will execute. This way the “for loop” iterates.

In “while loop”, the statements inside the loop executes until the condition is true.

```
while (condition){

//statements

}
```

In “do-while” loop, the condition is checked at the end of the loop. So, the loop executes at least once.

```
do{

//statements

} while(condition)
```

Program to find the factorialof 3 (3!) using iteration ("for loop") is as follows.

```
int main(){

intn=3, factorial=1;

inti;

for(i=1; i<=n ; i++){

factorial = factorial *  i;

}

printf(“Factorial is %d\n”, factorial);

return 0;

}
```

What are the Similarities Between Recursion and Iteration?

- Both are techniques to solve a problem.
- The task can be solved either in recursion or iteration.

What is the Difference Between Recursion and Iteration?

Recursion vs Iteration	
Recursion is a method of calling a function within the same function.	Iteration is a block of instructions which repeats until the given condition is true.
Space Complexity	
The space complexity of recursive programs is higher than iterations.	Space complexity is lower in iterations.
Speed	
Recursion execution is slow.	Normally, iteration is faster than recursion.
Condition	
If there is no termination condition, there can be an infinite recursion.	If the condition never becomes false, it will be an infinite iteration.
Stack	
In recursion, the stack is used to store local variables when the function is called.	In an iteration, the stack is not used.
Code Readability	
A recursive program is more readable.	The iterative program is harder to read than a recursive program.

Summary - Recursion vs Iteration

This article discussed the difference between recursion and iteration. Both can be used to solve programming problems. The difference between recursion and iteration is that recursion is a mechanism to call a function within the same function and iteration it to execute a set of instructions repeatedly until the given condition is true. If a problem can be solved in recursive form, it can also be solved using iterations.

Reference:

1.Point, Tutorials. “Data Structures and Algorithms Recursion Basics.”, Tutorials Point, 15 Aug. 2017. Available here
2.nareshtechnologies. "Recursion in C Functions | C Language Tutorial"YouTube, YouTube, 12 Sept. 2016. Available here
3.yusuf shakeel. "Recursion Algorithm | Factorial - step by step guide"YouTube, YouTube, 14 Oct. 2013. Available here

Image Courtesy:

1.'CPT-Recursion-Factorial-Code'By Pluke - Own work, (Public Domain) via Commons Wikimedia
2.'For-loop-diagram'By No machine readable author provided - Own work assumed. (CC BY-SA 2.5) via Commons Wikimedia

How to Cite this Article?

APA: Difference Between Recursion and Iteration .(2017 December 30). Retrieved (date), from http://differencebetween.com/difference-between-recursion-and-vs-iteration/

MLA: "Difference Between Recursion and Iteration" Difference Between.Com. 30 December 2017. Web.

Chicago: “Difference Between Recursion and Iteration ”. Difference Between.Com. http://differencebetween.com/difference-between-recursion-and-vs-iteration/accessed (accessed [date]).