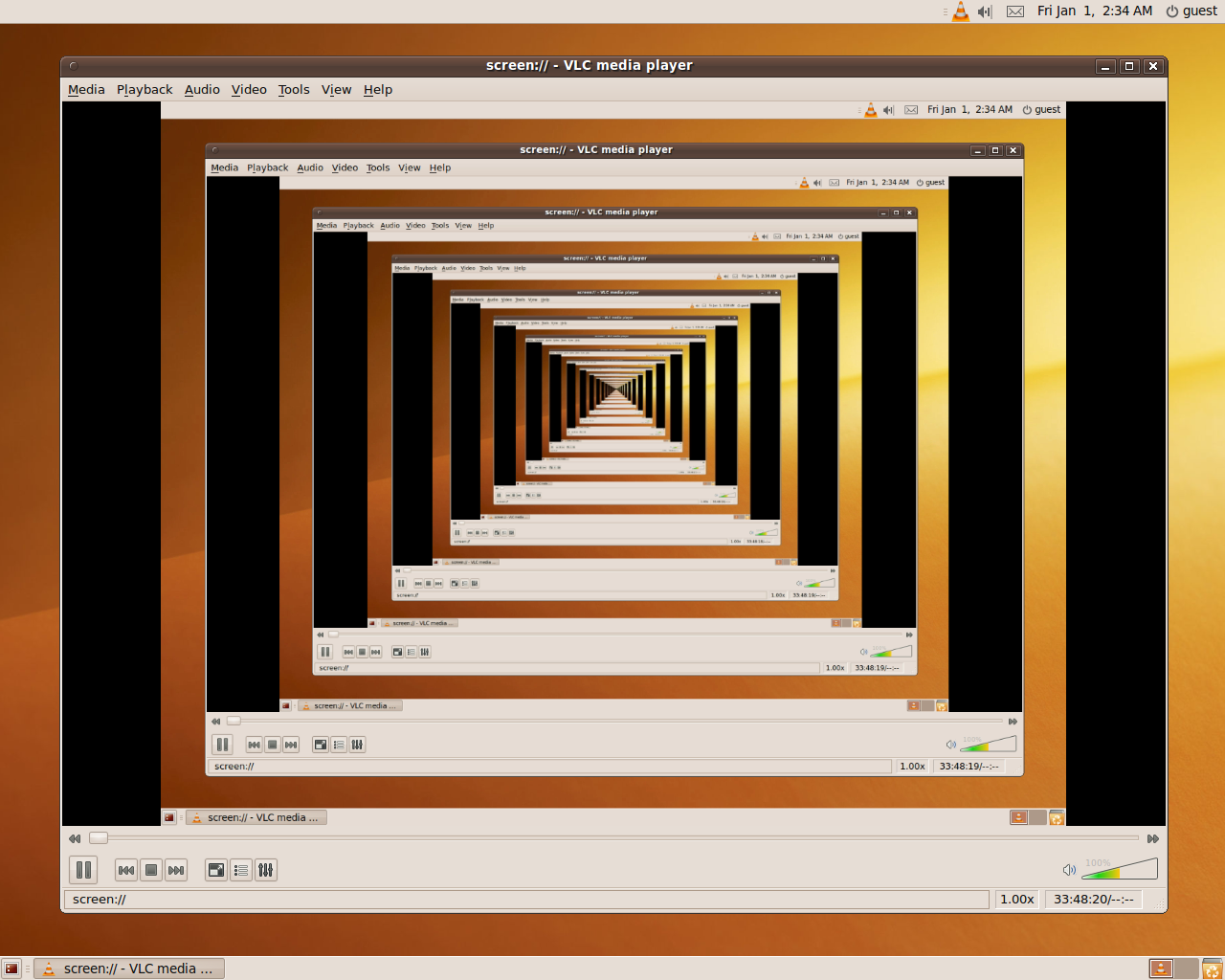
**রিকারশন/ Recursion , রিকার্সিভ অ্যালগরিদম, রিকার্সিভ ফাংশন ও সি প্রোগ্রামিং এ প্রয়োগ**

কোন কিছু যদি নিজেকে নিজে পুনরায় ডাকে, করে তাই হচ্ছে [**রিকারশন/ Recursion।**](http://en.wikipedia.org/wiki/Recursion_%28computer_science%29) যেমন একটা ছবির মধ্যে ঐ ছবিটি, ঐ ছবিটির ভেতরে আবার ঐ ছবিটি... এ প্রসেসটা হচ্ছে রিকারশনের একটা উদারহণ।

একটা খালি মাঠে গিয়ে নিজেকে নিজে ডাকলে বা একটা বিশাল হল রুমে নিজের নাম ধরে ডাকা ডাকি করলে রিকার্শন অনুভব করা যাবে।

নিচের ছবিটি দেখিঃ

[](http://tech.jakir.me/wp-content/uploads/2015/02/Screenshot_Recursion_via_vlc.png)

From Wikipedia

গুগলে Recursion লিখে সার্চ করে দেখুন। বার বার লেখা উঠবে Did you mean: Recursion। বানান ঠিক থাকার পর ও এটা দেখাবে। গুগল মজা করে একটা রিকারশন বসিয়ে দিয়েছে Recursion সার্চ টার্ম এর উপর।

**Recursive Algorithm** হচ্ছে যে অ্যালগরিদম নিজেকে নিজে কল করে, তা। কম্পিউটার প্রোগ্রামিং এ  রিকারসিভ অ্যালগরিদম ব্যবহার করে কোন প্রোগ্রামে রিকারশন ব্যবহার করা হয়। বিভিন্ন প্রোগ্রামিং ল্যাঙ্গুয়েজে একটি ফাংশন নিজেকে কল করার মাধ্যমে রিকারশন এর প্রয়োগ করা হয়। আমরা রিকার্সিভ ফাংশন ব্যবহার করে দুই একটা রিকার্সিভ অ্যালগরিদম দেখব। আর প্রোগ্রামিং ল্যাঙ্গুয়েজ হিসেবে ব্যবহার করব সি।

রিকার্সিভ ফাংশন কি তা তো এখন সহজেই বলা যাচ্ছে তাই না? যে ফাংশন নিজেকে নিজে কল করে, তাই হচ্ছে রিকার্সিভ ফাংশন। ফাংশন সম্পর্কে ধারণা না থাকলে এ লেখাটা দেখা যেতে পারেঃ[**সি তে ফাংশন**](http://jakir.me/c/functions/)

রিকারশনের সুবিধে হচ্ছে কোড সহজ করে, অনেক গুলো কোড লেখার পরিবর্তে কয়েক লাইনের কোড দিয়ে একটা সমস্যা সমাধান করা যায়।

***void f() {***  
***f() ...***  
***}***

এটা একটা রিকার্সিভ ফাংশন, কারণ ফাংশনটি নিজেকে নিজে কল করেছে। এটাকে বলে সরাসরি কল। আবার ফাংশন সরাসরি কল না করেও এমন একটা ফাংশনকে কল করতে পারে, যে ফাংশনটি প্রথম ফাংশনকে কল করে। নিচের উদারহণটি দেখিঃ

***void f() {***  
***g() ...***  
***}***

***void g() {***  
***f() ...***  
***}***

f ফাংশনটি g ফাংশনকে কল করেছে। আবার g ফাংশন f ফাংশনকে কল করেছে। এটাও রিকার্সিভ ফাংশন।

আমরা একটা রিকার্সিভ ফাংশন লিখি, যেটা ইউজার থেকে একটা নাম্বার নিবে, তারপর ঐ সংখ্যা থেকে এর পর ১ থেকে ঐ সংখ্যা পর্যন্ত সকল সংখ্যা প্রিন্ট করবে। এটা সিম্পল একটা প্রোগ্রাম। তবে আমরা এ প্রোগ্রামটি নতুন ভাবে লিখব। রিকারশন ব্যবহার করে। প্রোগ্রামটি যেহেতু সিম্পল, তাহলে আমরা একটু ভালো করে দেখলেই বুঝতে পারব। আর প্রোগ্রামটি কিভাবে কাজ করে, তা বুঝতে পারলেই আমরা রিকার্শন বুঝে ফেলব। এরপর আমরা কমপ্লেক্স সব প্রোগ্রাম রিকারশন ব্যবহার করে সহজেই লিখে ফেলতে পারব। আগে ফাংশনটি **Pseudo code [সুডো কোড]** এ আগে লিখি, এরপর সি প্রোগ্রামিং এ ইমপ্লিমেন্টেশন দেখব।

[**Pseudo code**](http://en.wikipedia.org/wiki/Pseudocode) হচ্ছে একটা প্রোগ্রাম বা একটা অ্যালগরিদমের ধাপ গুলো সাধারণ মানুষের ব্যবহার উপযোগি করে লেখা কিছু কোড। এগুলো কোন প্রোগ্রামিং ল্যাঙ্গুয়েজ ব্যবহার করে লেখা হয় না। এমন ভাবে লেখা হয় যেন মানুষ বুঝতে পারে। নিচে printInt নামে একটি ফাংশনের সুডো কোড দেওয়া হলো, যা ১ থেকে ঐ সংখ্যা পর্যন্ত সকল সংখ্যা প্রিন্ট করবে।

|  |
| --- |
| printInt( k ) { if (k == 0) { return 0; } print(k ); printInt( k - 1 );  } |

ফাংশনটি প্যারামিটার হিসেবে একটা ইন্টিজার নিবে। এরপর চেক করবে সংখ্যাটা কি ০? যদি শুন্য হয়, ঐখানেই ফাংশনের কাজ শেষ হবে। যদি ০ না হয় তাহলে ইন্টিজারটি প্রিন্ট করবে। এবং ঐ ইন্টিজার সংখ্যাটি থেকে ১ বিয়োগ করে আবার printInt কে কল করবে। মানে নিজেকে নিজে কল করবে।

এখন আমরা যদি printInt ফাংশনে 2 পাস করি, তাহলে ফাংশনটির তিনটে কপি তৈরি হবে। একটা হচ্ছে k এর মান 2 এর জন্য, একটা হচ্ছে k এর মান 1 এর জন্য। আর একটা হচ্ছে k এর মান ০ এর জন্য।

|  |  |  |
| --- | --- | --- |
| **যখন k এর মান 2** printInt( int k ) { if (k == 0) { return 0; } print(k ); printInt( k - 1 ); }  প্রিন্ট করবে 2 | **যখন k এর মান 1** printInt( int k ) { if (k == 0) { return 0; } print(k ); printInt( k - 1 ); } প্রিন্ট করবে 1 | **যখন k এর মান 0** printInt( int k ) { if (k == 0) { return 0; } print(k ); printInt( k - 1 ); } k এর মান ০ হওয়ায় কিছুই প্রিন্ট করবে না। |

এভাবে এখন আমরা যদি printInt ফাংশনে 5 পাস করি, তাহলে ফাংশনটির পাঁচটি কপি তৈরি হবে। অর্থাৎ যে সংখ্যা পাস করব, তত সংখ্যক বার ফাংশনটির কপি তৈরি হবে। আর এভাবেই রিকারশন কাজ করে।

এবার সি প্রোগ্রামিং এ  ইমপ্লিমেন্টেশন দেখিঃ

প্রোগ্রামটিতে প্রথমে আমরা আমাদের printInt ফাংশনটি লিখেছি। এর পর মেইন ফাংশনের ভেতর একটা ইন্টিজার ডিক্লেয়ার করেছি। তা ইউজার থেকে ইনপুট নিয়েছি। এরপর printInt ফাংশনে ইন্ট্রিজারটি পাস করেছি। আর printInt হচ্ছে রিকার্সিভ ফাংশন। যে নিজেকে নিজে কল করে আমাদের কাজ করে দিচ্ছে।

আরেকটা সিম্পল প্রোগ্রাম রিকারশন ব্যবহার করে লেখার চেষ্টা করি। যেমন একটা সংখ্যা ইউজার থেকে ইনপুট নিবে, তারপর ১ থেকে ঐ সংখ্যা পর্যন্ত সকল সংখ্যার যোগফল প্রিন্ট করবে। যখন ইনপুট হিসেবে 4 থাকবে তখন সাধারণ একটা প্রোগ্রাম যোগফল নির্নয় করবে নিচের মত করেঃ  
sum(4) = 1+2+3+4  
যখন ইনপুট হিসেবে থাকবে 5 তখনঃ  
sum(5) = 1+2+3+4+5  
যখন ইনপুট হিসেবে থাকবে 6 তখনঃ  
sum(6) = 1+2+3+4+5+6

যার মানে হচ্ছেঃ

|  |
| --- |
| sum(6) = sum(5) + 6 [sum(5)=(1+2+3+4+5)] sum(5) = sum(4) + 5 [sum(4)=(1+2+3+4)] sum(4) = sum(3) + 4 [sum(3)=(1+2+3)] sum(3) = sum(2) + 3 [sum(2)=(1+2)] sum(2) = sum(1) + 2 [sum(1)=(1)] sum(1) = sum(0) + 1 [sum(0)=(0)] sum(0) = 0 |

উপরের স্টেপ গুলো ৬টা সংখ্যার জন্য। এখন আমরা n তম সংখ্যার যোগফলের জন্য সহজ একটা ইকুয়েশন লিখে ফেলতে পারি, sum(n) = sum(n-1) + n.  
   
যখন ০ হবে, তখন প্রিন্ট করবে ০, আর যখন n হবে, তখন প্রিন্ট করবে sum(n) = sum(n-1) + n.

আর এটা সুডো কোডে লিখলেঃ

|  |
| --- |
| sum( int n ) { if( n == 0 ) return 0; else return n + sum( n-1 ); } |

সি পোগ্রামে এর প্রয়োগ করলেঃ

রিকার্শনের আরেকটা উদারহণ দেখি, এবার দেখব রিকার্শন ব্যবহার করে [Factorial](http://en.wikipedia.org/wiki/Factorial) বের করার উপায়।  
   
একটা সংখ্যার [**Factorial**](http://en.wikipedia.org/wiki/Factorial) হচ্ছে ১ থেকে ঐ সংখ্যা পর্যন্ত সব গুলো সংখ্যার গুণফল। একটা সংখ্যার ফ্যাক্টোরিয়াল এভাবে দেখানো হয়ঃ n! এখানে প্রথমেই ধরে নেওয়া হয় ০ এর Factorial হচ্ছে ১।

১ এর ফ্যাক্টোরিয়াল ১।  
দুই এর ফ্যাক্টোরিয়াল ১\*২ =২।  
৩ এর ফ্যাক্টোরিয়াল হচ্ছে ১\*২\*৩ = ৬।  
৪ এর ফ্যাক্টোরিয়াল হচ্ছে ১\*২\*৩\*৪ = ২৪।  
৫ এর ফ্যাক্টোরিয়ালঃ  
   
[factorial 5](http://tech.jakir.me/wp-content/uploads/2015/02/factorial-5.png)  
   
আমরা এভাবে লিখতে পারিঃ factorial(n) = (n \* factorial(n-1)); আর যেহেতু factorial(0) = 1, আমরা factorial এর জন্য একটা রিকার্সিভ ফাংশনের সুডো কোড লিখে ফেলতে পারিঃ

|  |
| --- |
| factorial(n) { if (n == 0) return 1; else return (n \* factorial(n-1)); } |

factorial(0) = 1

তাহলে যদি আমরা factorial(3) এর মান বের করি, ফাংশনটি এভাবে কাজ করবেঃ

|  |
| --- |
| factorial(3) = 3 \* factorial(2) 3 \* 2 \* factorial(1) 3 \* 2 \* 1 \* factorial(0) 3 \* 2 \* 1 \* 1 = 6 |

সি প্রোগ্রামিং এ ইমপ্লিমেন্টেশনঃ

এবার আপনি একটা সমস্যা রিকারশন ব্যবহার করে সলভ করতে চেষ্টা করুন। যেমন ইউজার থেকে একটা সংখ্যা ইনপুট নিবে, ধরি n। আর প্রোগ্রামটাকে করতে হবে কি, n তম Fibonacci সংখ্যাটা বের করতে হবে।  Fibonacci number কি, তা সম্পর্কে নিচের লিঙ্ক থেকে জানা যাবেঃ

[**Fibonacci number**](http://en.wikipedia.org/wiki/Fibonacci_number)