

## Tự học lập trình C - Bài 9: Đề qui

Chủ Nhật, 13/10/2013 08:42

### ĐỀ QUI

#### Giải thuật đệ quy:

Giải thuật đệ quy là giải thuật có chứa thao tác gọi đến chính nó. Giải thuật đệ quy cho phép mô tả một dãy lớn các thao tác bằng một số ít các thao tác trong đó có chứa thao tác gọi lại giải thuật (gọi đệ quy).

Giải thuật giải bài toán bằng đệ quy thường rất đẹp, gọn gàng, dễ hiểu, dễ sửa đổi. Tuy nhiên, việc xử lý giải thuật đệ quy lại thường gây khó khăn cho máy tính (tốn không gian nhớ và thời gian xử lý), hơn nữa không phải mọi ngôn ngữ lập trình đều cho phép mã hóa giải thuật đệ quy (ví dụ: FORTRAN).

#### Chương trình con đệ quy:

Chương trình con đệ quy là một chương trình con mà trong thân của nó có ít nhất một câu lệnh là lời gọi đến chính nó.

#### Chương trình con đệ quy phải có hai thành phần:

- Thành phần không chứa đệ qui, đó là điều kiện để kết thúc quá trình đệ qui.
- Thành phần có chứa đệ qui, sau mỗi bước, phạm vi của thành phần này phải thay đổi cho đến khi gặp điều kiện kết thúc.

**@Lưu ý: Muốn giải một bài toán bằng giải thuật đệ qui việc đầu tiên ta phải đưa bài toán về một dạng tổng quát. Từ đây ta phải đi xác định cho được điều kiện suy biến của bài toán (tức điều kiện để kết thúc giải thuật đệ qui) và điều kiện gọi đệ qui.**

#### Ví dụ bài toán tính n!

Ta có

```
n=0, 0!=1,
n=1, 1!=1x1 <=> 0!x1
n=2, 2!=1x1x2 <=> 1!x2
n=3, 3!=1x1x2x3 <=> 2!x3
=> n!=1x1x2x3x...x n <=> (n-1)! x n
```

Như vậy:

- Điều kiện suy biến khi n=0, 0!=1
  - Điều kiện gọi đệ qui n>0, n!=n x (n-1)!
- Vậy khi có được 0! => 1! => 2! => 3! ... => n!

Giải thuật tính n!

```
#include <stdio.h>

long int gthua(int n);

void main(void)
{
    int n;

    scanf("%d", &n);

    printf("Giá trị thừa của %d là: %d", n, gthua(n));
}

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

- Khi thực hiện lời gọi **gthua(3)** sẽ phát sinh lời gọi **gthua(2)**, đồng thời phải lưu giữ thông tin về trạng thái xử lý chưa hoàn thành (**return(3 \* gthua(2))**) vào Stack.

#### BÀI MỚI NHẤT

Lập trình với Python - Cú pháp cơ sở

Lập trình với Python - Bước đầu với Python

Disk Space Management of Operating System (3)

Disk Space Management of Operating System (2)

Disk Space Management of Operating System (1)

Năm lĩnh vực chức năng của công tác quản trị mạng (Computer Network Management)

Một số câu hỏi trắc nghiệm về Quản trị mạng và Giao thức SNMP

IBM bắt tay FPT phân phối giải pháp bảo mật mạng mới

MySQL – Bài 17: Import dữ liệu trong MySQL (1)

#### Java programming language



Lý thuyết cơ sở

Ngôn ngữ lập trình

Assembly

Pascal

C/C++

Java

Phát triển ứng dụng

LAMP & WAMP

Hệ điều hành

Tin học văn phòng

Điện toán đám mây

#### VIDEO MỚI NHẤT



#### BÀI XEM NHIỀU NHẤT



**Tự học lập trình C - Bài 1: Một số khái niệm cơ bản**  
Khái niệm tên rất quan trọng trong quá trình lập trình, ...



**MySQL – Bài 8: Khóa chính (primary key) và khóa ngoại (foreign key) của table**  
Với ràng buộc này thì, việc người sử dụng vô tình hay có ...



**Tự học lập trình C - Bài 10: Mảng một chiều**  
Mảng 1 chiều là tập hợp các phần tử có cùng kiểu dữ ...



**Tự học lập trình JAVA – Bài 1: Bước đầu với Java**  
Một chương trình java có thể được định nghĩa như là một ...



**Tự học lập trình C - Bài 2: Cấu trúc chương trình C**  
Một chương trình bao gồm một hoặc nhiều hàm, mỗi hàm ...



**Tự học lập trình Assembly - Bài 1: Bước đầu với lập trình Assembly trên vi xử lý Intel 8086/8088**  
Như đã biết, lệnh ngôn ngữ máy là một dãy các con số 0, ...



**Hướng dẫn in ấn trong Word 2007/2010 – Step by Step**  
[Tự học] - Tiêu đề đầu trang (Header)/tiêu đề cuối ...



#### CÂU HỎI MỚI

khác nhau giữa ngành an ninh mạng và quản trị mạng 0

Viết chương trình C/C++ kết nối với hợp ngữ 0

- Gặp lời gọi **gthua(2)**, tiếp tục làm phát sinh lời gọi **gthua(1)**, đồng thời vẫn phải lưu trữ thông tin về trạng thái xử lý còn dang dở (**return( 2\*gthua(1))**) vào Stack.
- Cứ như vậy cho tới khi gặp lời gọi của trường hợp suy biến (**return(1)**).
- Khi gặp trường hợp suy biến, những thông tin được lưu tạm trong Stack sẽ được lấy ra xử lý (thông tin lấy ra theo kiểu lưu trữ của Stack, thông tin vào sau sẽ được lấy ra trước). Và như vậy, dùng kết quả của gthua(0) để tính gthua(1), dùng kết quả của gthua(1) để tính gthua(2), dùng kết quả của gthua(2) để tính gthua(3). Cuối cùng được kết quả của phép tính giai thừa.

Cụ thể thực hiện lấy và tính toán trong Stack như sau:

- Lấy **return(1\*gthua(0))** để thực hiện **gthua(1)=1\*gthua(0)=1\*1=1**
- Lấy **return(2\*gthua(1))** để thực hiện **gthua(2)=2\*gthua(1)=2\*1=3**
- Lấy **return(3\*gthua(2))** để thực hiện **gthua(3)=3\*gthua(2)=3\*2=6**

#### Bài tập thực hành

1. Sử dụng đệ qui để viết hàm tìm ước số chung lớn nhất của 2 số
2. Sử dụng đệ qui để viết hàm tính tổng  $S = 1+2+...+n$ .



#### Bài cùng chủ đề

- Tự học lập trình C - Bài 10: Mảng một chiều - 13/10/2013
- Tự học lập trình C - Bài 8: Chương trình con - Hàm - 19/08/2013
- Tự học lập trình C - Bài 7: Câu lệnh lặp (2) - 15/08/2013
- Tự học lập trình C - Bài 6: Câu lệnh lặp (1) - 15/08/2013
- Tự học lập trình C - Bài 5: Câu lệnh điều kiện - 07/08/2013
- Tự học lập trình C - Bài 4: Nhập/xuất dữ liệu - 02/08/2013
- Tự học lập trình C - Bài 3: Hằng - Biến - Toán tử - B.thức - 01/08/2013
- Tự học lập trình C - Bài 2: Cấu trúc chương trình C - 01/08/2013
- Tự học lập trình C - Bài 1: Một số khái niệm cơ bản - 01/08/2013

#### Bài mới hơn

- Tự học lập trình C - Bài 11: Mảng nhiều chiều - 09/01/2014
- Tự học lập trình C - Bài 10: Mảng một chiều - 13/10/2013

#### Các bài khác

- Tự học lập trình C - Bài 8: Chương trình con - Hàm - 19/08/2013
- Tự học lập trình C - Bài 7: Câu lệnh lặp (2) - 15/08/2013
- Tự học lập trình C - Bài 6: Câu lệnh lặp (1) - 15/08/2013
- Tự học lập trình C - Bài 5: Câu lệnh điều kiện - 07/08/2013
- Tự học lập trình C - Bài 4: Nhập/xuất dữ liệu - 02/08/2013
- Tự học lập trình C - Bài 3: Hằng - Biến - Toán tử - B.thức - 01/08/2013
- Tự học lập trình C - Bài 2: Cấu trúc chương trình C - 01/08/2013
- Tự học lập trình C - Bài 1: Một số khái niệm cơ bản - 01/08/2013

#### Ý kiến của bạn

Họ tên (Yêu cầu)

E-mail (Yêu cầu, nhưng sẽ không hiển thị)



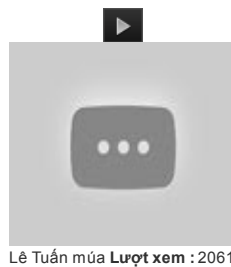
1000 symbols left

☐ Thông báo cho tôi khi có bình luận tiếp theo



Làm mới

Gửi



Lê Tuấn mùa Lượt xem : 2061

#### VIDEO NỔI BẬT



Bài 21: Hướng dẫn tự học theo ví dụ mẫu Lượt xem : 16471



vẽ một cung tròn trong góc phần tư thứ nhất với các thông số sau đây được nhập từ bàn phím: toạ độ tâm, bán kính, góc ở tâm. ai giúp e với ạ. e đang cần gấp ạ. gmail e là: vtmy95@gmail.com. thanks ạ!!!

xin giúp đỡ về Hợp Ngữ asm

0

Viết chương trình C/C++ kết nối với hợp ngữ assembly. giúp em với ạ. ibx: em vào gmail: dodangduong.it.1992@gmail.com

0

em tạo 1 class trong java, ví dụ : class hang\_hoa{}, vậy em có thể tạo ra từ lớp này số lượng đối tượng được nhập vào từ bên ngoài chưa biết trước không?

0

ac em có thể tìm tài liệu ấn bản java mobile(tiếng việt) trên trang web nào ko.thanks

0

Với mô hình mạng sau, có thể share mạng được không?

0

AIA

0



Thích Trang Chia sẻ

Hãy là người đầu tiên trong số bạn bè của bạn thích nội dung này



<a href="#">Kiến thức An ninh mạng</a>	<a href="#">Quản trị mạng</a>	<a href="#">Giải pháp mạng</a>	<a href="#">Tin bảo mật</a>	<a href="#">Hệ thống - Thiết bị</a>	<a href="#">Lý thuyết cơ sở</a>
<a href="#">Lý thuyết nền tảng</a>	<a href="#">Window s Server</a>	<a href="#">Hạ tầng mạng</a>	<a href="#">Bảo mật Desktop/Server</a>	<a href="#">Desktop - Server</a>	<a href="#">Cơ sở dữ liệu</a>
<a href="#">Kiến trúc bảo mật</a>	<a href="#">Linux/Ubuntu Server</a>	<a href="#">Dịch vụ mạng</a>	<a href="#">Bảo mật Netw ork/Service</a>	<a href="#">LAN - WAN</a>	<a href="#">Nguyên lý Hệ điều hành</a>
<a href="#">An ninh E-commerce</a>	<a href="#">Netw ork Devices</a>	<a href="#">Bảo mật mạng</a>	<a href="#">Bảo mật Mobile Devices</a>	<a href="#">Security Devices</a>	<a href="#">Mạng máy tính</a>
<a href="#">An ninh Cloud Computing</a>	<a href="#">Other Tools</a>	<a href="#">Hệ thống E-commerce</a>	<a href="#">Kiến thức IT</a>	<a href="#">Other Devices</a>	<a href="#">Nội dung khác</a>
<a href="#">Ngôn ngữ lập trình</a>	<a href="#">Phát triển ứng dụng</a>	<a href="#">LAMP &amp; WAMP</a>	<a href="#">Hệ điều hành</a>	<a href="#">Tin học văn phòng</a>	<a href="#">Tin học Phổ thông</a>
<a href="#">Assembly</a>	<a href="#">Ứng dụng Desktop</a>	<a href="#">SQL &amp; MySQL</a>	<a href="#">MS Window s</a>	<a href="#">Microsoft Office</a>	<a href="#">Tin học 10</a>
<a href="#">Pascal</a>	<a href="#">Ứng dụng Web</a>	<a href="#">Web Server</a>	<a href="#">Linux - Ubuntu</a>	<a href="#">Open Office</a>	<a href="#">Tin học 11</a>
<a href="#">C/C++</a>	<a href="#">Ứng dụng Mobile</a>	<a href="#">PHP - Pert - Python</a>	<a href="#">Mac OS</a>	<a href="#">Mẹo dùng Office</a>	<a href="#">Tin học 12</a>
<a href="#">Java</a>	<a href="#">Game Mobile</a>	<a href="#">phpMyAdmin</a>	<a href="#">Mobile OS</a>	<a href="#">Excel cho Kế toán</a>	<a href="#">Olympic Tin học</a>

Copyright © 2013 Tự Học An Ninh Mạng. All Rights Reserved.

Vui lòng ghi rõ nguồn "[Tuhocanninhmang.com](http://Tuhocanninhmang.com)" khi phát hành lại thông tin từ website này.
[Trang chủ](#)
[Tìm kiếm](#)
[Sitemap](#)
[Liên hệ](#)
[User](#)