TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
MÔN: **KĨ THUẬT LẬP TRÌNH**

HƯỚNG DẪN THỰC HÀNH

TUẦN 04

KĨ THUẬT ĐỆ QUY

TP.HCM, ngày 10 tháng 05 năm 2020

MỤC LỤC

1 Giới thiệu

Một hàm được gọi là đệ quy nếu bên trong than của hàm đó có lời gọi hàm lại chính nó một cách tường minh hay tiềm ẩn.

2 Phân loại

Có bốn loại đệ quy trong kỹ thuật lập trình: đệ quy tuyến tính, đệ quy nhị phân, đệ quy hỗ tương, đệ quy phi tuyến. Trong đó, đệ quy phi tuyến là trường hợp tổng quát của đệ quy tuyến tính và đệ qui nhị phân, còn đệ quy nhị phân lại là trường hợp tổng quát của đệ quy tuyến tính.

2.1 Đệ quy tuyến tính

Một hàm được gọi là đệ quy tuyến tính nếu bên trong than hàm có duy nhất một lời gọi hàm đến chính nó.

Cấu trúc hàm như sau

Dòng	
1	KDL Ten_ham(Danh sách tham số){
2	if(<Điều kiện dừng>){
3	…
4	return <giá trị trả về>
5	}
6	...
7	... Ten_ham(Danh sách tham số);
8	...
9	}

Ví dụ minh họa: Viết hàm đệ quy tính tổng S(n) = 1 + 2 + … + n

Ta thấy S(n – 1) = 1 + 2 + … + n – 1. Vậy ta có S(n) = S(n – 1) + n (Với S(0) = 0)

Cuối cùng ta có thể viết hàm đệ quy như sau:

Dòng	
1	long Tong(int n){
2	if(n == 0) return 0;
3	return n + Tong(n – 1);
4	}
5	}

2.2 Đệ quy nhị phân

Một hàm được gọi là đệ quy nhị phân nếu trong than của hàm đó có hai lời gọi hàm gọi lại chính nó.

Cấu trúc hàm như sau:

Dòng	
1	KDL Ten_ham(Danh sách tham số){
2	if(<Điều kiện dừng>){
3	…
4	return <giá trị trả về>
5	}
6	...
7	... Ten_ham(Danh sách tham số);
8	...
9	... Ten_ham(Danh sách tham số);
10	...
11	}

Ví dụ minh họa: Viết hàm đệ quy tính số hạng thứ n của dãy Fibo.

$S(n) = 1 + 1 \times 2 + 1 \times 2 \times 3 + \ldots + 1 \times 2 \times \ldots \times n$

Ta thấy $S(n-1) = 1 + 1 \times 2 + \ldots + 1 \times 2 \times \ldots \times (n-1)$

$S(n-2) = 1 + 1 \times 2 + \ldots + 1 \times 2 \times \ldots \times (n-2)$

Vậy $S(n) = S(n-1) + n!$ (1) và $S(n-1) = S(n-2) + (n-1)!$ (2). Nhân 2 về của 2 cho n ta có $n \times S(n-1) = n \times S(n-2) + n! \Rightarrow n! = n \times S(n-1) - n \times S(n-2)$ (3). Thay (3) và (1) ta có $\mathbf{S(n) = S(n-1) + n \times S(n-1) - n \times S(n-2) = (n+1) \times S(n-1) - n \times S(n-2)}$.

Cuối cùng ta có thể viết hàm đệ quy như sau:

Dòng	
1	long Tong(int n){
2	if(n == 0) return 0;
3	if(n == 1) return 1;
4	return (n + 1)*Tong(n – 1) - n*Tong(n – 2);
5	}

2.3 Đệ quy hỗ tương

Hai hàm được gọi là đệ quy hỗ tương nếu trong than của hàm này có lời gọi hàm tới hàm kia và bên trong than hàm kia có lời gọi hàm tới hàm này.

Cấu trúc hàm như sau

Dòng	
1	KDL Ten_ham_1(Danh sách tham số){
2	if(<Điều kiện dừng>){
3	…
4	return <giá trị trả về>
5	}
6	...
7	... Ten_ham_2(Danh sách tham số);
8	...
9	}
10	
11	KDL Ten_ham_2(Danh sách tham số){
12	if(<Điều kiện dừng>){
13	…
14	return <giá trị trả về>
15	}
16	...
17	... Ten_ham_1(Danh sách tham số);
18	...
19	}

Ví dụ minh họa: Giả sử ta có hai hàm BaoLoi và InChuoi, nếu chuỗi vượt quá ngưỡng MAX_LEN thì sẽ báo lỗi.

Dòng	
1	#include <stdio.h>
2	#include <string.h>
3	#define MAX_LEN 50
4	
5	void BaoLoi();
6	void InChuoi(char* ch);
7	
8	void InChuoi(char* ch){
9	if(strlen(ch) <= MAX_LEN) printf("%s\n", ch);
10	else BaoLoi();
11	}
12	

```
13  void BaoLoi(){
14    InChuoi("Chuoi vuot qua chieu dai qui dinh");
15  }
```

2.4 Đệ qui phi tuyến

Hàm được gọi là đệ qui phi tuyến khi trong than hàm có lời gọi hàm lại chính nó được đặt bên trong than vòng lặp.

Cấu trúc hàm như sau

```
Dòng
1   KDL Ten_ham(Danh sách tham số){
2     if(<Điều kiện dừng>){
3       …
4       return <giá trị trả về>
5     }
6     for(int i = 0; i < n; i++){
7     {
8       …
9       Ten_ham(Danh sách tham số);
10      …
11    }
12  }
```

Ví dụ minh họa: viết hàm tính số hạng thứ n của dãy

$$\begin{cases} x(0) = 1 \\ x(n) = n \times x(n-1) + (n-1) \times x(n-2) + (n-2) \times x(n-3) + \ldots + 1 \times x(0) \end{cases}$$

Ta nhận thấy chuỗi x(n) hoàn toàn phụ thuộc vào n chuỗi phía sau, và tương tự chuỗi x(n – 1) cũng phụ thuộc vào n – 1 chuỗi phía sau của nó và tương tự các chuỗi còn lại. Ta có mã nguồn bên dưới như sau:

```
Dòng
1   long Tong(int n){
2     if( n == 0) return 1;
3     long s = 0;
4     for(int i = 1; i < n; i++) s += i*Tong(i – 1);
5     return s;
6   }
```

3 Bài tập

Sinh viên xây dựng các hàm sau theo phong cách đệ quy

- Tính: $S(n) = 1 \times 2 \times \ldots \times n$

- Tính: $S(n) = 1 + (1/3) + (1/5) + \ldots + 1/(2n+1)$

- Tính: $S(n) = (1/2) + (3/4) + \ldots + ((2n+1)/(2n+2))$

- Tính: $S(n, x) = x + x^2 + x^3 + \ldots + x^n$

- Tính: $S(n, x) = x + (x^2/(1 + 2)) + (x^3/(1 + 2 + 3)) + \ldots + (x^n/(1 + 2 + \ldots + n))$

- Đếm số lượng số nguyên tố trong mảng một chiều các số nguyên dương

- Đếm số lượng số hoàn thiện trong mảng một chiều các số nguyên dương

- Tính tổng các số chẵn trong mảng một chiều các số nguyên dương

- Liệt kê vị trí ở đó các phần tử là nguyên tố trong mảng một chiều các số nguyên dương

- Liệt kê vị trí ở đó các phần tử là chẵn trong mảng một chiều các số nguyên dương

- Tính tổng các giá trị lớn hơn giá trị đứng liền trước nó trong mảng một chiều các số thực

- Đếm phân biệt các giá trị trong mảng một chiều.

Sinh viên tự thiết kế hàm main minh họa **TẤT CẢ** các **hàm chính** bên trên.