Chào mừng các bạn trở lại với Series Lập Trình Căn Bản C. Hôm nay chúng ta sẽ tiếp tục bài học về Mảng trong ngôn ngữ C.

Ở những bài trước, ta đã biết về Mảng và một vài kỹ thuật liên quan, bây giờ chúng ta sẽ tìm hiểu về cách thao tác Hàm trong Mảng.

**Mảng thao tác với hàm**

Khi hàm thao tác với các biến bình thường, chúng ta sẽ truyền vào nó những tham số bình thường.

Khi chúng ta muốn truyền vào mảng, cách truyền của chúng tương tự như biến bình thường và thêm dấu **[].**

Ví dụ: Truyền vào theo kiểu biến.

int TinhTong(int a, int b)

{

// tính tổng và trả về giá trị

}

Ví dụ 2: Truyền vào theo kiểu mảng

int TinhTong(int a[], int n) // đưa vào mảng a và n phần tử

{

int i;

for ( i = 0 ; i < n ; i++)

{

// thao tác tính tổng

}

}

Tương tự như vậy, chúng ta có thể viết hàm nhập mảng

void NhapMang(int a[], int n) // đưa vào mảng a và n phần tử

{

int i;

for ( i = 0 ; i < n ; i++)

{

// thao tác nhập mảng

printf("nhap a[%d]: ",i);

scanf("%d", &a[i]);

}

}

Hoặc thao tác xuất mảng cũng tương tự.

Và khi các bạn học về Hàm rồi, các bạn sẽ được học thêm một vài kỹ thuật mới.

1. Kỹ thuật cờ hiệu ( cắm cờ)

Kỹ thuật cắm cờ thường được sử dụng để kiểm tra sự đồng bộ của mảng.

VD: Kiểm tra một mảng toàn chẵn, một mảng toàn lẽ.

Kiểm tra mảng có số nguyên tố không.

Kiểm tra mảng có gì đó bị sai không....

Mô tả kĩ thuật:

B1: Cho một biến để kiểm tra, ban đầu cho giá trị là đúng.

VD int check = 1; // 1 nghĩa là đúng

B2: Trong quá trình xử lí, cho kiểm tra xem có giá trị nào không thỏa với yêu

cầu bài toán hay không. Nếu có thì đưa biến ban đầu thành 0 ( mang giá trị sai).

Và ngừng kiểm tra.

B3: Kiểm tra biến ban đầu và kết luận.

VD một bài toán kiểm tra số toàn chẵn

int i;

int check = 1; // biến kiểm tra

for ( i = 0; i < n ; i++)

{

if ( a[i] %2 == 1) // kiểm tra xem có số nào là lẽ hay không

{

// Có thì đưa check = false và thoát vòng lặp

check = 0 ;

break;

}

}

// kiểm tra biến và kết luận

if ( check == 0)

printf("khong phai toan chan");

else

printf("mang toan chan");

Hay một ví dụ về kiểm tra một mảng có phải toàn số nguyên tố hay không

int i;

int check = 1; // biến kiểm tra

for ( i = 0; i < n ; i++)

{

if (Check\_SNT(a[i]) == 0) // kiểm tra xem có số nào không phải SNT hay không

{

// Có thì đưa check = false và thoát vòng lặp

check = 0 ;

break;

}

}

// kiểm tra biến và kết luận

if ( check == 0)

printf("khong phai toan SNT");

else

printf("mang toan SNT");

1. Kĩ thuật lính canh.

Đây là kĩ thuật thường để tìm thằng nào đó NHẤT với điều kiện gì đó.

VD: giá trị lớn nhất trong mảng, bé nhất trong mảng....

Mô tả thuật toán:

B1: Bạn lấy giá trị ban đầu làm nhất.

VD: max = a[0]; // phần tử a[0] sẽ là lớn nhất.

B2: Bạn duyệt mảng và kiểm tra xem có thằng nào lớn hơn nó không.

Nếu có thì đưa nó làm max. Không thì tiếp tục duyệt.

B3: In ra kết quả

VD về hàm tìm số lớn nhất trong mảng.

int max= a[0]; // gán phần tử a[0] cho max ban đầu

for ( i = 1 ; i < n ; i++) // duyệt mảng, không cần kiểm tra thằng đầu tiên vì mình đã gán

{

if ( a[i] > max ) // kiểm tra xem có thằng nào lớn hơn max hay không

{

// gán max = nó nếu có.

max = a[i];

}

}

3. Kĩ thuật sắp xếp nổi bọt (bubblesort)

Như tên gọi, nó sẽ giúp chúng ta sắp xếp lại mảng theo chiều tăng dần hoặc giảm dần.

Mô tả thuật toán.

Thuật toán sử này sử dụng 2 vòng lặp và có độ phức tạp là O(N2

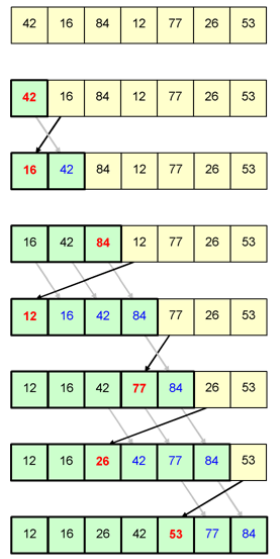
)

B1: Lồng 2 vòng lặp lại ( giả sử là i và j).

Lấy i = 0 và j = i +1;

B2: Ta duyệt vòng lặp lần lượt với mỗi thằng i ta so sánh với j thằng.

Hơi khó hiểu. Vì vậy chúng ta hãy xem ảnh, nó sẽ giúp chúng ta hiểu hơn.



Code demo

for ( i=0 ; i<n-1 ;i++) // vòng lặp i bắt đầu từ 0 tới n - 2 ( bé hơn n - 1 )

{

for( j= i+1 ; j<n ;j++) // vòng lặp j bắt đầu từ i + 1 tới n -1 ( bé hơn n)

{

if( a[i] > a[j] ) // nếu có a[i] mà lớn hơn a[j] thì hoán vị

{

// hoán vị a[i] với a[j]

}

}

}

Bài học về mảng của chúng ta đến đây là kết thúc, phần sau chúng ta sẽ học về chuỗi.

Bài tập: Phần này sẽ là bài tập tổng hợp của 3 phần, nên có thể nhiều, các bạn có thể bỏ qua một số bài nếu đã làm rồi.

Bài 1. Nhập, xuất mảng, tính tổng các phần tử của mản

Bài 2. Đảo ngược nội dung của mảng. Ví dụ mảng a = { 9 8 7 6 } ◊ a = { 6 7 8 9 }.

Bài 3. Xóa một phần tử của mảng tại vị trí vt. Ví dụ: mảng a có n = 5 phần tử, a = { 9 8 7 3 5 }. Xóa tại vị trí vt = 2 ◊ mảng a = { 9 8 3 5 } (n = 4)

Bài 4. Chèn thêm một phần tử vào vị trí vt. Ví dụ: mảng a có n = 5 phần tử, a = { 9 8 7 3 5 }. Chèn phần tử -4 ở vị trí vt = 2 ◊ mảng a = { 9 8 -4 7 3 5 } (n = 6).

Bài 5. Nhập vào giá trị x. Cho biết giá trị x có nằm trong mảng a hay không. Xuất ra vị trí xuất hiện trong mảng a (nếu có). Chỉ cần xuất ra vị trí đầu tiên. Ví dụ a = { 9 8 7 0 10 7 }.

• Nếu x = 7 thì x có trong mảng a (a[2] = 7 ◊ vị trí là 2).

• Nếu x = 0 ◊ vị trí là 3.

• Nếu x = 4 thì x không có trong mảng.

Bài 5. Kiểm tra xem mảng có chứa toàn số lẻ hay không. Bạn làm bài này bằng 3 cách:

• Cách 1: đếm số lượng số lẻ trong mảng.

• Cách 2: sử dụng kỹ thuật đặt cờ hiệu.

• Cách 3: cho vòng lặp duyệt mảng và return kết quả ngay nếu a[i] là số chẵn.

Bài 6: Viết chương trình nhập vào mảng nguyên chứa N phần tử, kiểm tra xem mảng đó có bao nhiêu số Nguyên tố.