

Bài thực hành mảng cơ bản

Bài 1:

a). Viết định nghĩa cho các hàm có khai báo:

(1) `void nhap(int A[], int &N);`, hàm này cho phép nhập số N phần tử của mảng, sau đó tiếp tục nhập vào nội dung (từng giá trị một) của mảng A ứng với số phần tử ấy.

(2) `void xuat(int A[], int N);`, hàm này cho phép in ra N giá trị đang được lưu trữ trong mảng A.

b). Viết hàm `main()` trong đó khai báo một mảng A các số nguyên có MAX phần tử (tự định trị sao cho thỏa đáng) và một số nguyên N cho biết số số nguyên đang được dùng của mảng A. Sau đó gọi hàm `nhap()` để nhập vào các giá trị cho A, rồi gọi hàm `xuat()` để xuất ra nội dung của mảng và xem có đúng là nội dung mà bạn đã nhập vào hay không.

Lưu ý: Hai hàm `nhap()` và `xuat()` này sẽ được dùng lại trong các bài tập sau.

Bài 2:

a). Viết định nghĩa cho các hàm có khai báo:

(1) `void xuatchan(int A[], int N);`, có chức năng in ra tất cả các số chẵn của mảng A.

(2) `void xuatVTchan(int A[], int N);`, có chức năng in ra tất cả các số ở vị trí chẵn của mảng A.

(3) `void thanhle(int A[], int N);`, có chức năng cộng những số chẵn của mảng A với 1, nghĩa là sau khi thực hiện, ta được mảng A gồm toàn những số lẻ.

b). Viết hàm `main()` trong đó khai báo mảng nguyên A có MAX phần tử và một số nguyên N chỉ số phần tử thực tế của mảng này. Gọi hàm `nhap()` để nhập nội dung cho mảng này, sau đó lần lượt gọi các hàm `xuatchan()`, `xuatVTchan()`, sau cùng là hàm `thanhle()` rồi hàm `xuat()` để kiểm tra xem liệu mảng A đã được biến đổi thành mảng chứa số lẻ chưa.

Bài 3:

a). Viết định nghĩa cho các hàm có khai báo sau:

(1) `int timNhoNhat(int A[], int N);`, tìm và trả về giá trị nhỏ nhất của dãy số nguyên A có N phần tử.

(2) `double trungbinh(int A[], int N);`, có chức năng tính giá trị trung bình của các phần tử của mảng A có N phần tử.

(3) `int timkiem(int A[], int N, int k);`, có chức năng tìm xem trong mảng A gồm N phần tử có phần tử nào có giá trị bằng với k không; nếu có thì trả về vị trí ấy, nếu không có thì trả về -1.

b). Viết hàm `main()` với các khai báo mảng như bài 10, gọi hàm `nhap()` và các hàm vừa định nghĩa ở câu a). để xem giá trị nhỏ nhất, trị trung bình của mảng A. Sau đó khai báo và cho nhập giá trị của biến k, gọi hàm `timkiem()` và in ra thông báo là có tìm được số k trong mảng A hay không.

Bài 4: Dùng lại hàm `timkiem()` ở bài 11.a.(3). Viết chương trình cho đọc vào các số nguyên có giá trị trong khoảng 10 đến 100. Khi mỗi số được đọc vào, hãy in ra nếu số này chưa từng xuất hiện trước đó. Chương trình cho in ra 20 giá trị số nguyên khác nhau.

Bài 5: Viết hàm cho nhận vào một mảng các số nguyên và số phần tử của mảng. Hàm có chức năng kiểm tra xem đây có phải là mảng đối xứng hay không? Nếu có đối xứng thì trả về 1, nếu không thì trả về 0.

Ví dụ:

Mảng đối xứng: [2 4 5 7 5 4 2]

Mảng không đối xứng: [2 3 8 7 4]

Bài 6: Viết hàm có các chức năng sau:

a). Tính và lưu N số Fibonacci đầu tiên vào một mảng nguyên A.

b). Tính tổng của nghịch đảo N-1 số Fibonacci đầu tiên, không dùng phần tử F_0 .

c). Đảo ngược nội dung của mảng A.

Ví dụ: Với $N = 7$, mảng A là:

A = [0 1 1 2 3 5 8 13]

Sau khi đảo ngược, A trở thành:

A = [13 8 5 3 2 1 1 0]

Bài 7: Sắp xếp chọn (Selection Sort) một dãy số là cách sắp xếp mà ở mỗi bước ta tìm phần tử nhỏ nhất trong dãy rồi phần tử này được hoán chuyển với phần tử đầu dãy (nghĩa là đưa phần tử nhỏ nhất về đầu dãy).

Ví dụ: Dãy ban đầu: 6 5 1 3 → Dãy hoán đổi: 1 5 6 3

Sau đó lặp lại bước này với dãy con được hình thành bằng dãy trước đó bỏ đi phần tử đầu tiên.

Ví dụ: Ở bước 2, dãy ban đầu: 5 6 3 → Dãy hoán đổi: 3 6 5

Việc lặp lại này kết thúc khi dãy con chỉ có 1 phần tử. Khi đó, ta được toàn bộ dãy đã được sắp xếp.

Hãy viết hàm nhận vào một mảng các số nguyên và số phần tử của mảng đó, sau đó sắp xếp mảng số nguyên này thành dãy tăng dần dùng phương pháp Selection Sort.

Home work:

Các hàm liên quan đến mảng:

a. Sắp xếp mảng tăng/giảm dần.

b. Tìm vị trí lần đầu tiên của số x trong mảng.

c. Tìm tất cả các vị trí xuất hiện của x, kết quả lưu trong mảng phụ.

d. Xóa phần tử tại vị trí index trong mảng.

e. Xóa tất cả các phần tử bằng x trong mảng.

f. Xóa tất cả các phần tử trùng nhau trong mảng.

g. Xóa các phần tử trùng nhau, chỉ chừa lại một phần tử lần đầu tiên. Ví dụ: trong mảng có 3 phần tử có giá trị là 10, xóa 2 phần tử phía sau, chỉ để lại một phần tử lần đầu tiên có giá trị bằng 10 mà thôi.

h. Chèn một số x vào mảng tại vị trí index.

i. Chèn x vào mảng A có thứ tự sao cho mảng sau khi chèn vẫn có thứ tự.

j. Trộn 2 mảng A, B có thứ tự thành mảng C có thứ tự.

k. Tách mảng A thành 2 mảng con B, C sao cho B chứa các số chẵn, C chứa các số lẻ.

