

1. Yêu cầu.

Hai mảng giống nhau là hai mảng có cùng số phần tử và các phần tử ở vị trí giống nhau là như nhau.

Viết chương trình nhận vào từ bàn phím hai mảng và trả lời hai mảng có giống nhau không.

Input

- Dòng đầu tiên chứa số nguyên n
- Hai dòng tiếp theo lần lượt là n
- n số nguyên thuộc hai mảng a và mảng b

Output

- Nếu hai mảng giống nhau không, in ra "YES", ngược lại in ra "NO".

Gợi ý

1. Cần duyệt mảng từng vị trí.
2. Nếu có một vị trí khác nhau, in ra "NO" và dừng chương trình
3. Nếu duyệt hết mảng mà không tìm ra vị trí nào khác nhau, in ra "YES" và dừng chương trình

2. Điểm tổng kết Lập trình nâng cao của n sinh viên được lưu trong một mảng 1 chiều.

Yêu cầu

Viết chương trình sắp xếp lại bảng điểm trên theo thứ tự giảm dần và in ra màn hình.

Input

- Dòng đầu chứa số nguyên n
- Dòng thứ hai chứa n số thực là điểm của n sinh viên

Output

- Một dòng chứa n điểm số đã được sắp xếp theo thứ tự giảm dần, hai số liền nhau cách nhau bằng một dấu cách
- Các con số được làm tròn đến chữ số thập phân thứ hai

Gợi ý

1. Để sắp xếp ta sử dụng hai vòng lặp lồng nhau để duyệt mảng

2. Vòng lặp thứ nhất chỉ số i từ 0 đến $n-1$
3. Vòng lặp thứ hai chỉ số j từ $i+1$ đến $n-1$, tức là $j > i$
4. Trao đổi hai phần tử thứ i và thứ j nếu chúng chưa được sắp xếp giảm dần

3. Yêu cầu:

Viết chương trình nhận đầu vào là số nguyên n và dãy gồm n số nguyên đã được sắp xếp theo thứ tự tăng dần, in ra màn hình dãy với các số riêng biệt sắp xếp theo thứ tự tăng dần (tức là loại bỏ các số lặp lại nhiều lần trong dãy ban đầu).

Input:

- Dòng đầu tiên là độ dài n , $0 < n \leq 1000$.
- Dòng thứ 2 chứa n số đã được sắp xếp theo thứ tự tăng dần cách nhau bởi dấu cách

Output:

- In ra màn hình dãy với các số không trùng nhau theo thứ tự tăng dần

Gợi ý:

1. Khởi tạo biến chứa giá trị n và nhập giá trị n .
2. khởi tạo mảng, dùng vòng for để nhập mảng.
3. Dùng vòng for để duyệt giá trị của mảng. In ra số tại vị trí i nếu $i == 0$ hoặc số đó khác số ở vị trí trước đó.

4. Yêu cầu

Cho một mảng gồm n số thực đã được sắp xếp theo thứ tự tăng dần.

Viết chương trình chèn một số thực k vào mảng trên sao cho thứ tự sắp xếp không đổi. In dãy mới ra màn hình. Kết quả làm tròn đến số thập phân thứ hai.

Input:

- Dòng đầu tiên là độ dài n , $0 < n \leq 1000$.
- Dòng thứ 2 chứa n số thực cách nhau bởi dấu cách.
- Dòng thứ 3 chứa số thực x cần chèn vào.

Output:

- Dãy $n+1$ số thực đã được chèn thêm số mới với thứ tự sắp xếp tăng dần và được làm tròn đến số thập phân thứ 2

Gợi ý:

1. Khai báo biến số nguyên để lưu trữ số lượng phần tử và nhập dữ liệu.
2. Khởi tạo mảng, dùng vòng lặp để nhập vào từng phần tử của mảng.
3. Khai báo số thực cần chèn vào và nhập dữ liệu.
4. Khai báo biến chạy i và j
5. Xác định vị trí chèn số thực, dùng vòng `for` và biến chạy i để duyệt mảng từ đầu đến vị trí đầu tiên lớn hơn số thực x . Dùng lệnh `break`; để dừng vòng `for`.
6. Các số từ vị trí i đến cuối mảng cần được đẩy lùi về sau 1 vị trí. Dùng vòng `for` với biến chạy j từ cuối mảng đến vị trí i : `array[j+1] = array[j]`;
7. Cuối cùng gán vị trí phần tử i bằng x

5. Yêu cầu

Yêu cầu

Một mảng số nguyên có độ dài n được cho là đối xứng nếu phần tử thứ k trong mảng

($0 \leq k \leq n-1$) có giá trị bằng phần tử thứ $n-1-k$.

Viết chương trình nhập vào mảng số nguyên có n phần tử, kiểm tra xem mảng đó có đối xứng hay không.

Nếu có, in ra màn hình "Symmetric array.", ngược lại, in ra màn hình "Asymmetric array."

Input

- Hai dòng văn bản, dòng 1 chứa số $n > 0$ là kích thước của mảng số nguyên.
- Dòng 2 chứa n số nguyên phân tách bởi dấu cách là n phần tử của mảng.

Output

- Nếu mảng đối xứng in ra "Symmetric array.", ngược lại, in ra màn hình "Asymmetric array."

Gợi ý

1. Đọc dữ liệu vào biến n .
2. Khai báo mảng số nguyên có n phần tử, dùng vòng `for` để đọc dữ liệu vào mảng.

3. Cách 1: for 1 chiều => dùng công thức để so sánh số đối diện.
4. Cách 2: Dùng 2 biến chạy đầu cuối