

Lưu ý: Bài quy đổi ra 2 slot short hoặc 1 slot Long

Title

Sắp xếp mảng một chiều với thuật toán sắp xếp nổi bọt (bubble sort algorithm).

Background

N/A

Program Specifications

Thiết kế một chương trình cho yêu cầu đầu vào

- Số mục (phần tử) của mảng một chiều.
- Giá trị của các mục trong mảng phải là số nguyên.

Menu của chương trình như sau:

1. Các mục đầu vào của mảng
2. Sắp xếp mảng theo thứ tự tăng dần
3. Sắp xếp mảng theo thứ tự giảm dần
4. Thoát

Chi tiết hàm:

Hàm 1: Hiển thị menu và yêu cầu người dùng chọn một tùy chọn.

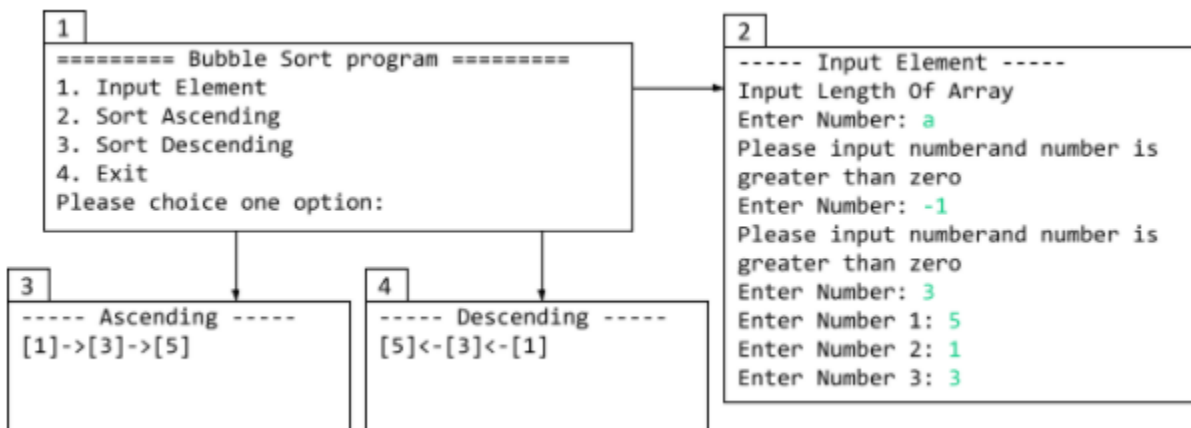
- Người dùng chạy chương trình. Chương trình nhắc người dùng chọn một tùy chọn.
- Người dùng chọn một tùy chọn, thực hiện **Hàm 2**.

Hàm 2: Thực hiện chức năng dựa trên tùy chọn đã chọn.

- Lựa chọn 1: (Các) thuộc tính đầu vào.
 - a. Yêu cầu đối với thông tin tác vụ đầu vào bao gồm "độ dài của mảng, số".
 - b. Kiểm tra dữ liệu hợp lệ với các điều kiện sau:
 - i. Độ dài của mảng phải lớn hơn 0.
 - c. Lưu (các) phần tử của mảng.
 - d. Quay lại màn hình chính.

- Lựa chọn 2: Sắp xếp theo thứ tự tăng dần
 - a. Hiển thị phần tử của mảng theo thứ tự tăng dần
 - b. Quay lại màn hình chính
- Lựa chọn 3: Sắp xếp theo thứ tự giảm dần
 - a. Hiển thị phần tử của mảng theo thứ tự giảm dần
 - b. Quay lại màn hình chính
- Lựa chọn 4: Thoát khỏi chương trình.

Kỳ vọng giao diện người dùng:



Gợi ý:

Thực hiện phương thức:

checkIn

sortAscending

sortDescending

Bubble sort là một thuật toán sắp xếp đơn giản lặp đi lặp lại các bước qua danh sách được sắp xếp, so sánh từng cặp mục liền kề và hoán đổi chúng nếu chúng không đúng thứ tự. Việc chuyển qua danh sách được lặp lại cho đến khi không cần hoán đổi, điều này cho biết rằng danh sách đã

được sắp xếp. Thuật toán này là một sắp xếp so sánh, được đặt tên theo cách các phần tử nhỏ hơn "bong bóng" lên đầu danh sách.

Định dạng 1: Sắp xếp từ đầu đến cuối. Bắt đầu từ đầu hàng, so sánh và thay đổi vị trí của từng cặp đến cuối hàng để di chuyển các phần tử lớn nhất về cuối hàng. Sau đó kiểm tra các phần tử còn lại trong hàng và lặp lại quá trình sắp xếp.

Định dạng 2: Sắp xếp từ cuối đến đầu. Bắt đầu từ cuối hàng & thay đổi vị trí của từng cặp để di chuyển phần tử nhỏ hơn của cặp phần tử đến vị trí bắt đầu của hàng hiện có. Sau đó, không kiểm tra nó trong bước tiếp theo. Lặp lại các bước trên cho đến khi không còn cặp nào.

Hàm 1: Khởi tạo mảng và nhập các mục của mảng.

- Chương trình sẽ hỏi về số lượng phần tử và yêu cầu nhập phần tử.
- Kiểm tra đầu vào thông qua hàm "public integer **checkIn** (String inputVal)".
- Thực hiện hàm: public Integer **checkIn**(String inputVal).

inputVal : Giá trị đầu vào

Giá trị trả về: số hoặc rỗng

Hàm 2: Sắp xếp theo mảng tăng dần.

- Chương trình thay đổi vị trí của các mục nên kết quả là một mảng chứa các phần tử được sắp xếp theo thứ tự tăng dần.
- Thực hiện hàm: int[] **sortAscending**(int[] arrayNeedSort)

- Đầu vào:

arrayNeedSort: mảng cần được sắp xếp

- Giá trị trả về: là mảng chứa các phần tử được sắp xếp theo thứ tự tăng dần

Hàm 3: Sắp xếp theo mảng giảm dần

- Chương trình thay đổi vị trí của các phần tử tạo ra một mảng chứa các phần tử được sắp xếp theo thứ tự giảm dần
- Thực hiện hàm: int[] **sortDescending**(int[] arrayNeedSort)

- Đầu vào:

arrayNeedSort: mảng cần được sắp xếp

- Giá trị trả về: là mảng chứa các phần tử được sắp xếp theo thứ tự giảm dần