

Câu 1 (2 điểm):

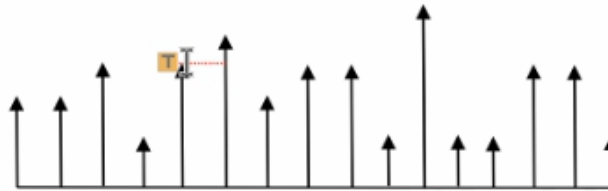
Năm 938 trận chiến trên sông bạch đằng của Ngô Quyền với quân Nam Hán luôn là một chiến tích quan trọng mà ai trong chúng ta đều ghi nhớ. Lấy cảm hứng từ trận thắng này, bạn được giao nhiệm vụ thiết kế cọc nhọn trong một trò chơi sắp tới, các cọc nhọn được mô tả như sau:

Mỗi cọc nhọn sẽ có chiều cao từ 1 m trở lên, cho bạn danh sách các cọc nhọn được cắm liên tiếp nhau, cách nhau 1 m theo một hàng ngang. Tính từ đáy của sông, các cọc này sẽ tạo thành rãnh với chiều cao khác nhau, rãnh được tạo ra bởi 2 cọc liên tiếp nhau. Bạn hãy tìm rãnh có chiều cao cao nhất.

Ví dụ: Cho danh sách các cọc có độ cao sau khi cắm xuống sông như sau:

[2, 2, 3, 1, 3, 4, 2, 3, 3, 1, 5, 1, 1, 3, 3, 1]

Hình vẽ bên dưới minh họa các cọc được cắm dưới sông. Rãnh có chiều cao cao nhất là 3, rãnh được tô màu đỏ. Có thể có nhiều rãnh có độ cao bằng 3 bạn chỉ cần đưa 1 đáp án rãnh cao nhất là được.



Câu 2 (4 điểm):

Xây dựng một danh sách liên kết các số nguyên có n phần tử, viết chương trình xử lý các yêu cầu sau:

- **Yêu cầu 1:** Kiểm tra nếu có các số chẵn trong danh sách liên kết mà nằm gần nhau thì thay đổi vị trí để không có bất kỳ 2 số chẵn nào nằm kế nhau. Không dùng mảng phụ.
- **Yêu cầu 2:** Bắt đầu từ vị trí Head, tìm số nguyên tố gần nhất Head (gần nhất bên phải) đem giá trị đó về đứng sau vị trí Head. Tương tự đến phần tử tiếp theo tìm số nguyên tố gần nhất và kéo về đứng sau giá trị tiếp theo này, lần lượt làm như vậy cho đến khi đi hết danh sách liên kết.

Ví dụ: 4 (Head) -> 8 -> 3 -> 6 -> 11 -> 13 -> 2 (Tail)

Danh sách liên kết sau khi biến đổi:

4 (Head) -> 3 -> 8 -> 11 -> 6 -> 13 -> 2 (Tail)

Lưu ý: Thiết kế Menu cho người dùng chọn các chức năng. Tất cả đều thao tác trên danh sách liên kết, **không** dùng mảng phụ, đổ dữ liệu từ danh sách liên kết vào mảng rồi xử lý, sau đó bỏ lại vào danh sách liên kết.

Câu 3 (4 điểm):

Cho một ma trận vuông có $N \times N$ phần tử, mỗi phần tử chứa một số thực, bây giờ bạn cần đi ma trận theo hình xoắn ốc bắt đầu từ vị trí dòng 0, cột 0 (phần tử tô màu đậm) để lần lượt tạo thành các nhóm trong danh sách liên kết theo quy tắc sau:

- Các nhóm đi từ trái sang phải hoặc từ phải sang trái, danh sách liên kết được tạo ra sẽ thuận theo chiều đi của hình xoắn ốc.
- Các nhóm đi từ trên xuống dưới hoặc từ dưới lên trên, các số sẽ in ngược lại để tạo thành một nhóm mới (màu đỏ) trong ví dụ bên dưới.

Khi đi qua tất cả các phần tử chương trình sẽ tạo thành một danh sách liên kết, theo như hình mẫu ví dụ bên dưới.

Ví dụ:

3	1	6	7.1
8	4	5	2
1	9	2	9
4	2	10	12

Kết quả sau khi đi theo hình xoắn ốc theo mô tả đề bài:

3 -> 1 -> 6 -> 7.1 -> 12 -> 9 -> 2 -> 10 -> 2 -> 4 -> 8 -> 1 -> 4 -> 5 -> 2 -> 9

- Yêu cầu 1:** Đếm xem có bao nhiêu nhóm được tạo trong danh sách liên kết khi đi ma trận theo quy tắc trên. Trong ví dụ trên sẽ có tổng cộng 7 nhóm.
- Yêu cầu 2:** In danh sách liên kết ra đúng định dạng như mô tả của đề bài. Trong ví dụ in ra đúng theo thứ tự đã mô tả ở trên.