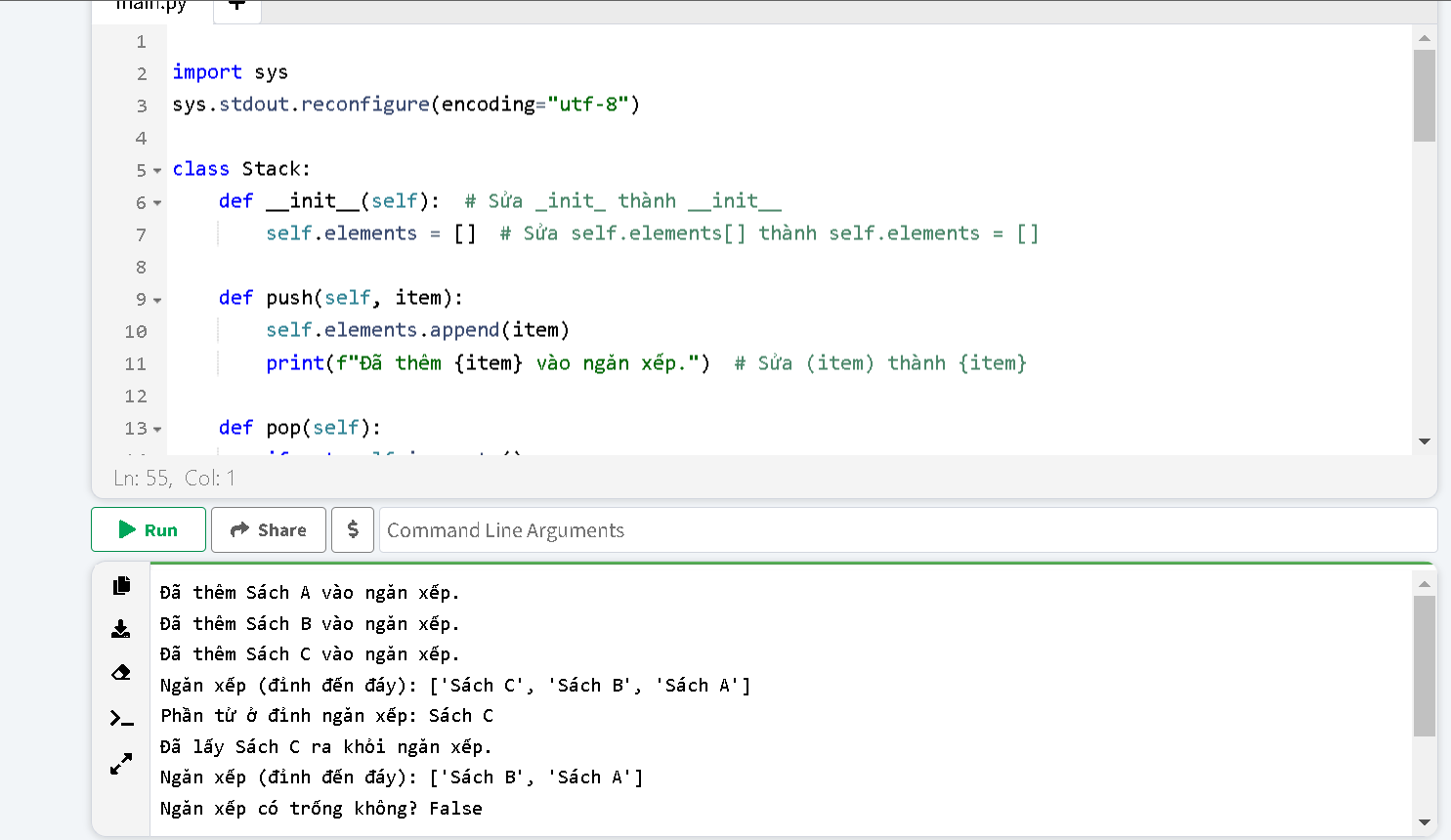
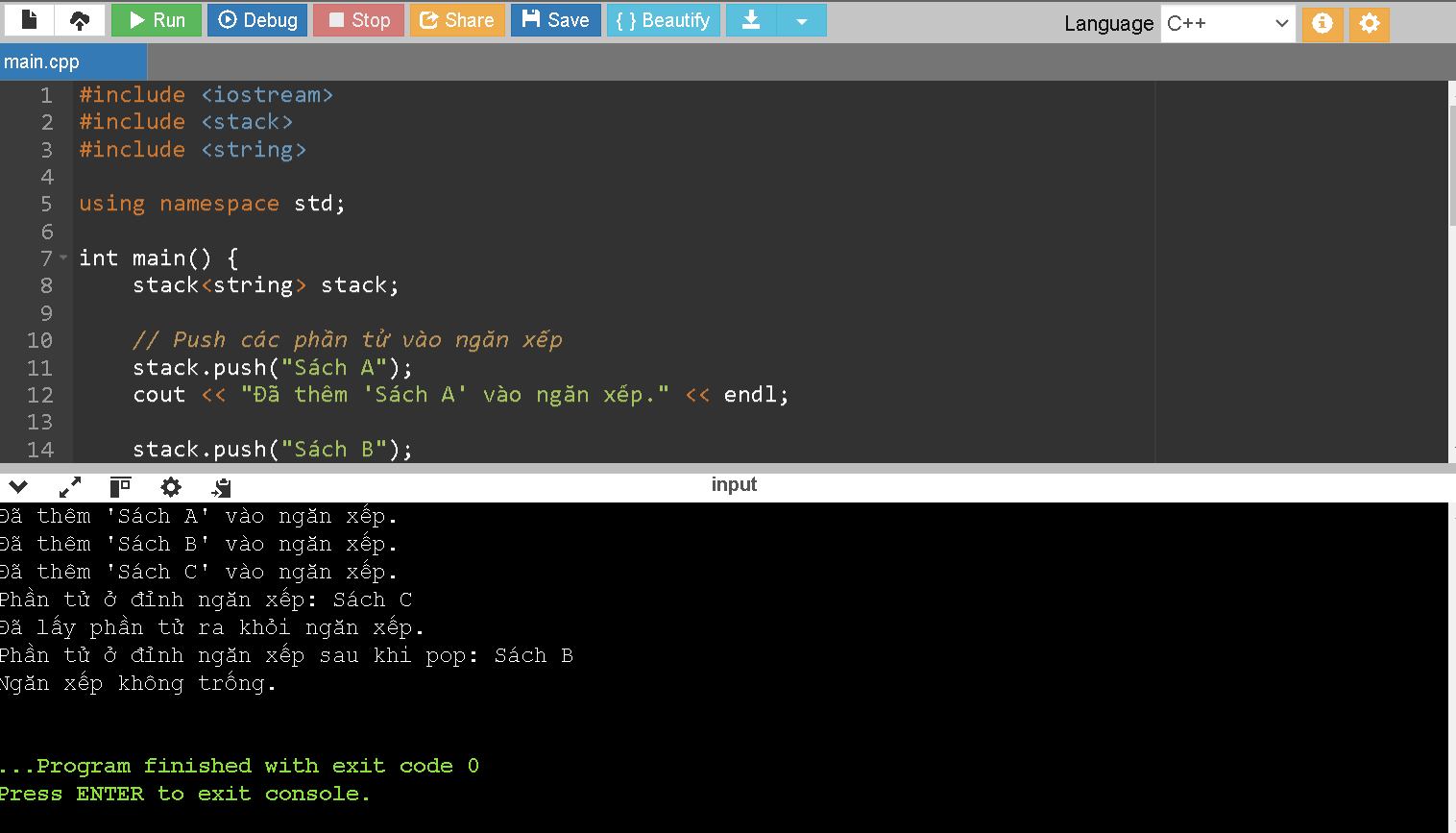
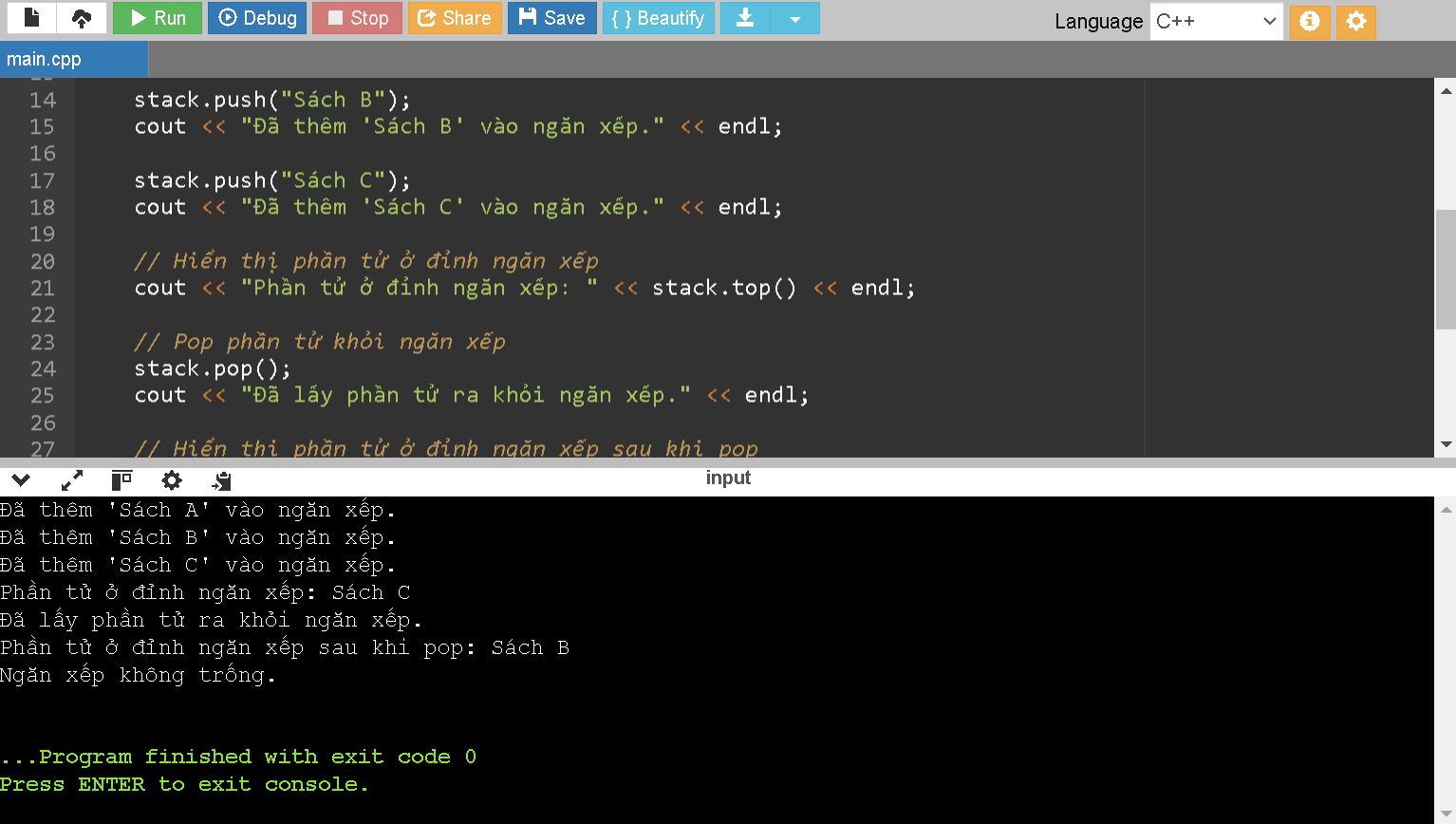
1.4.1



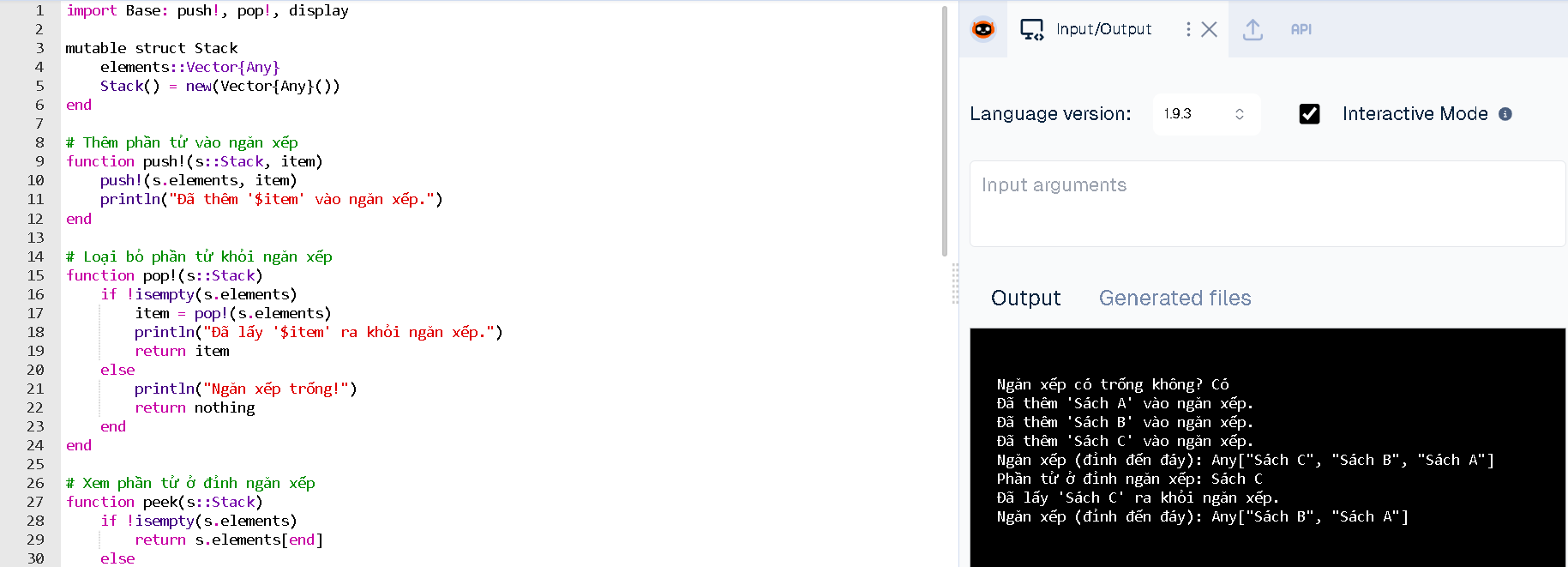
1.4.2



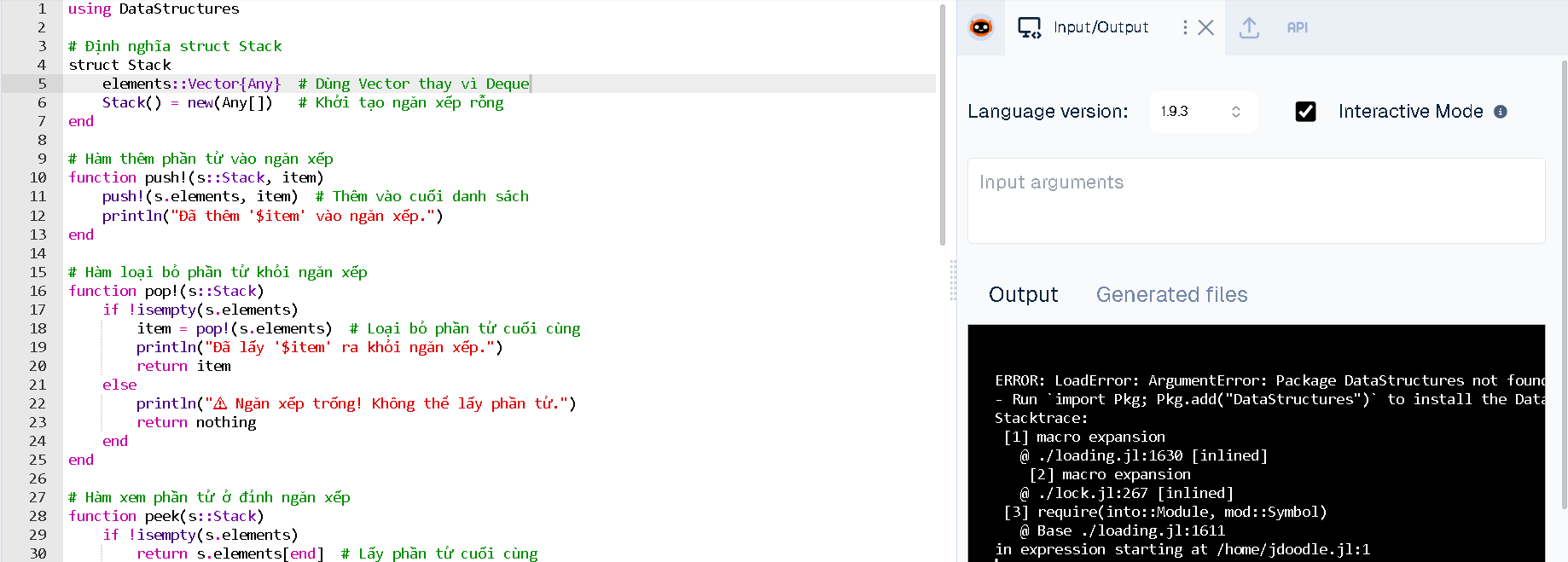
1.4.2.1



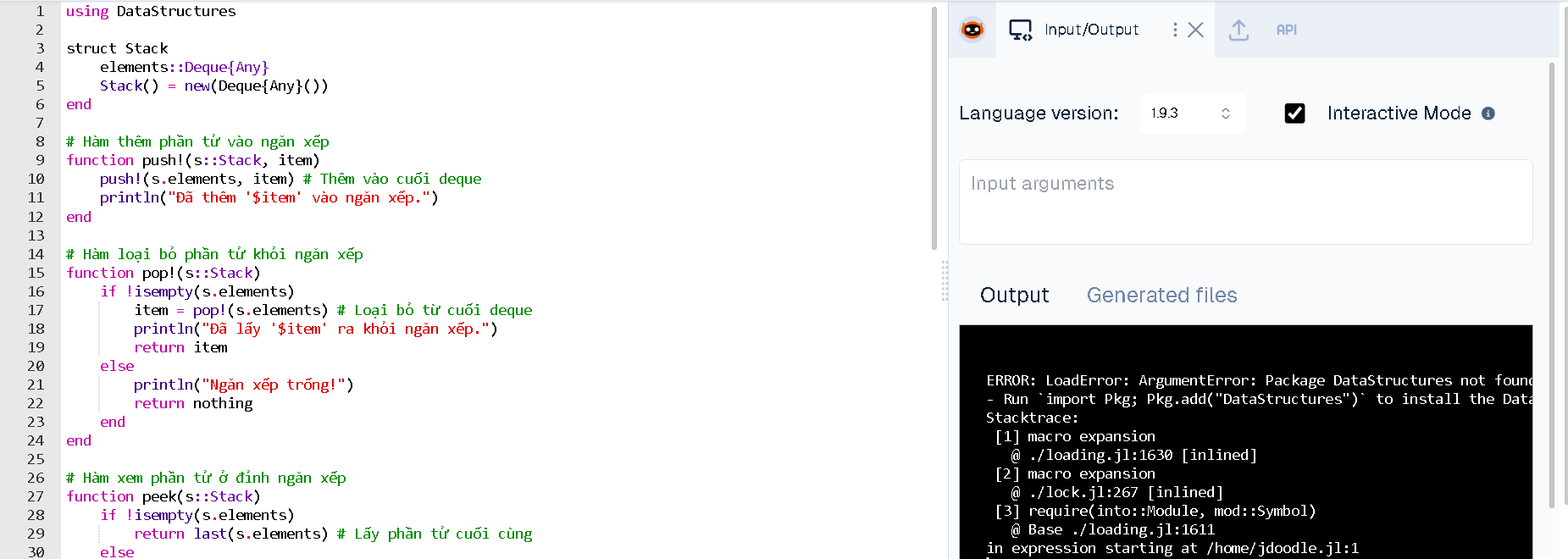
1.4.3



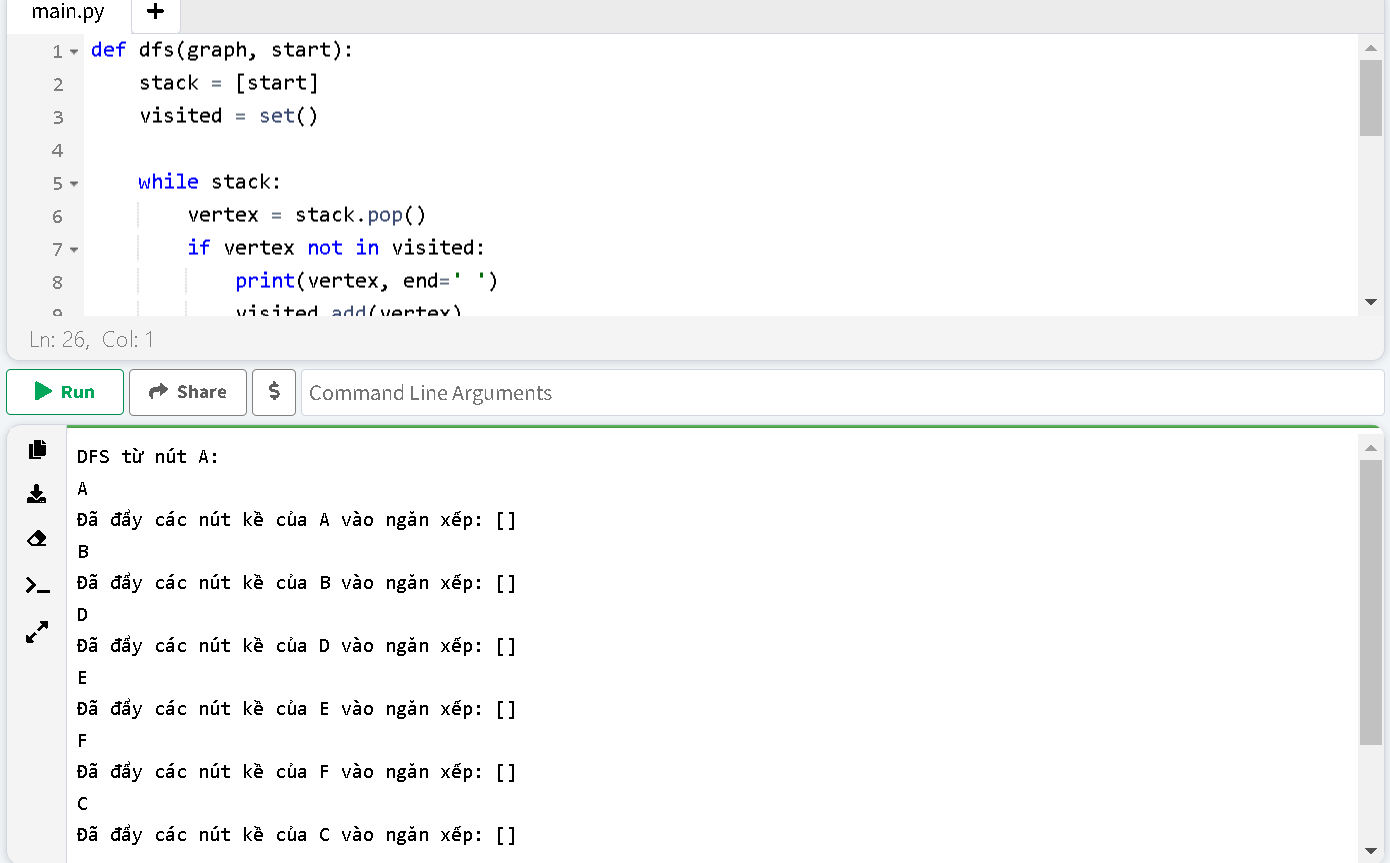
1.4.3.1



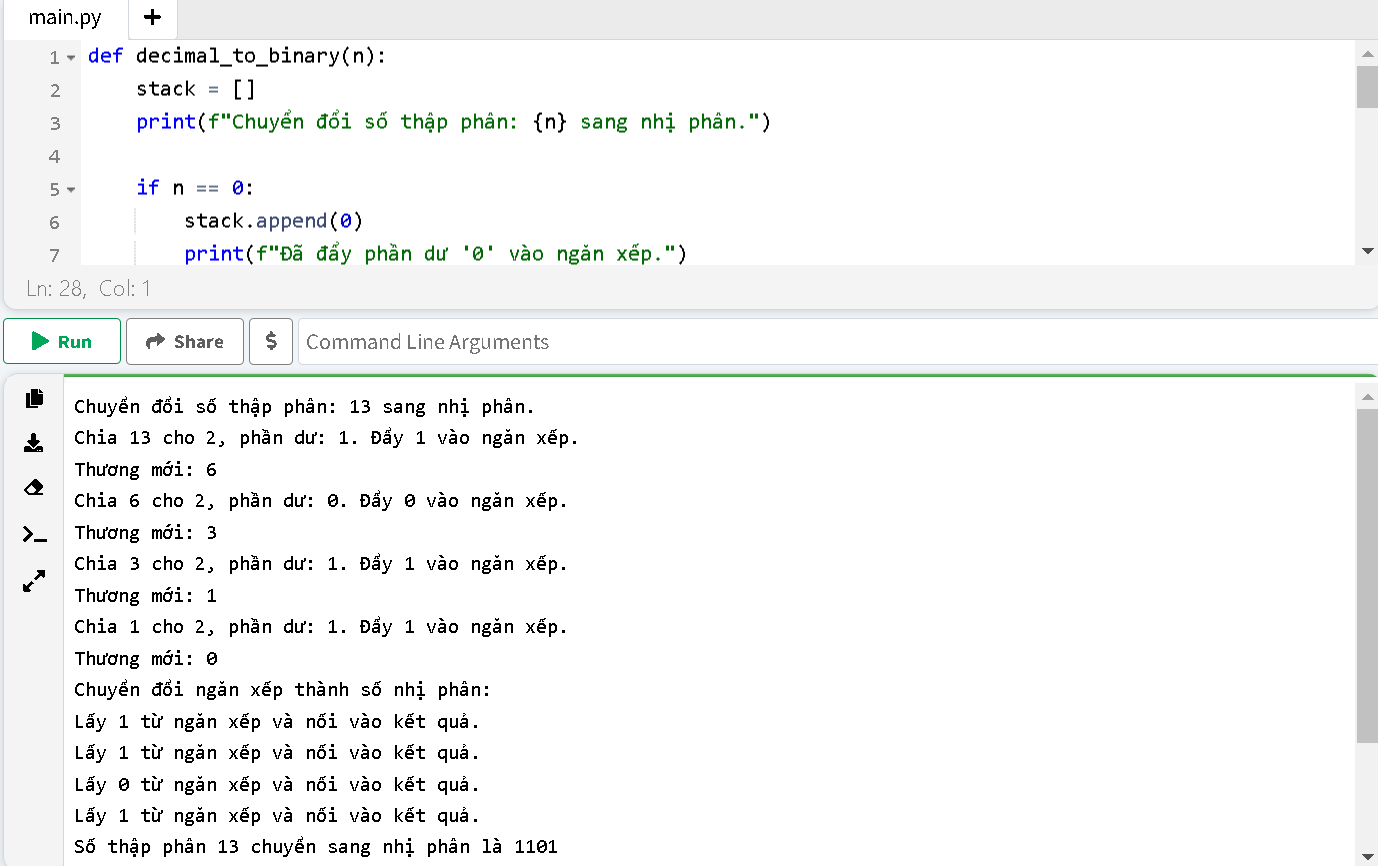
1.4.3.2



1.6.2

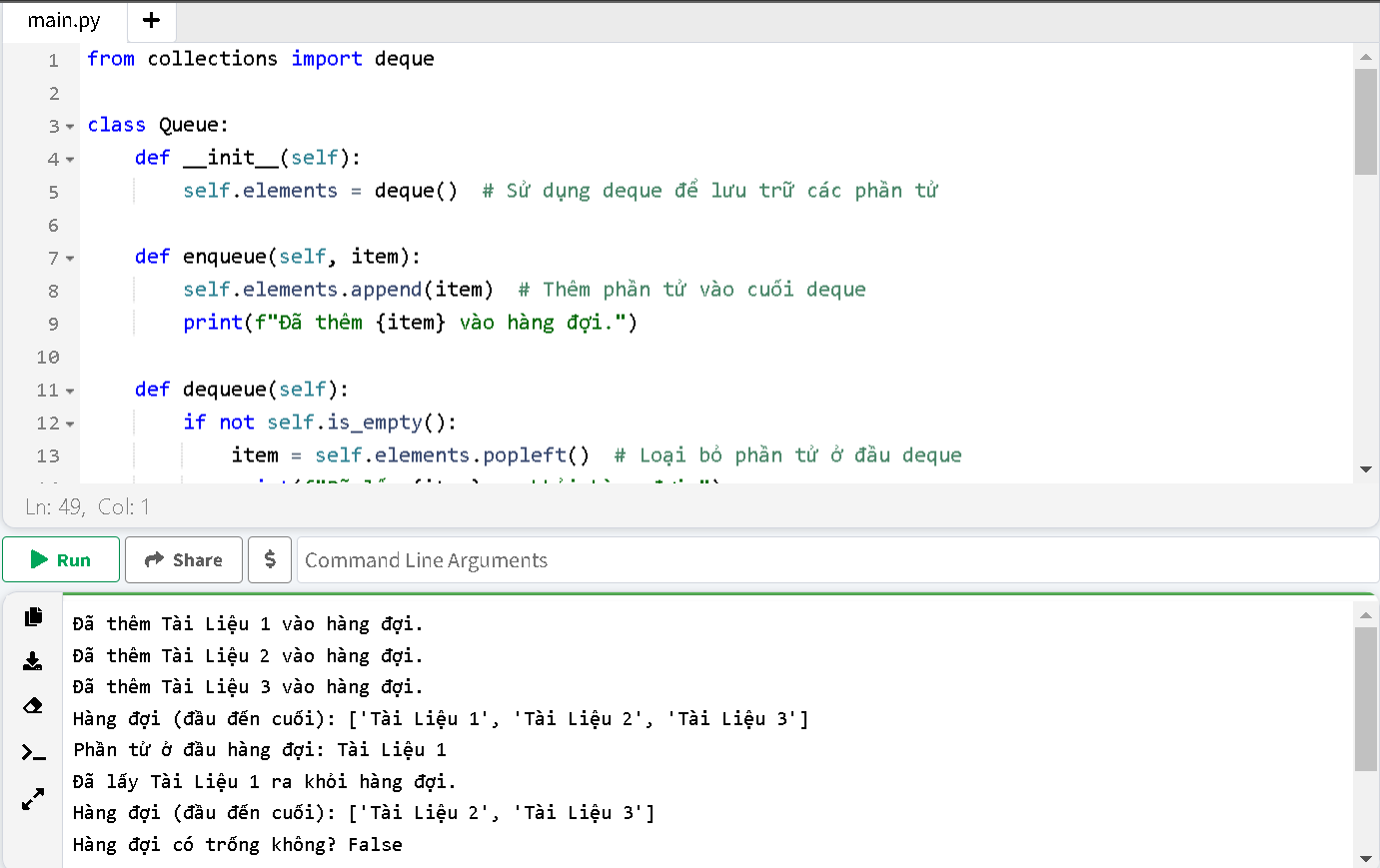


1.6.3

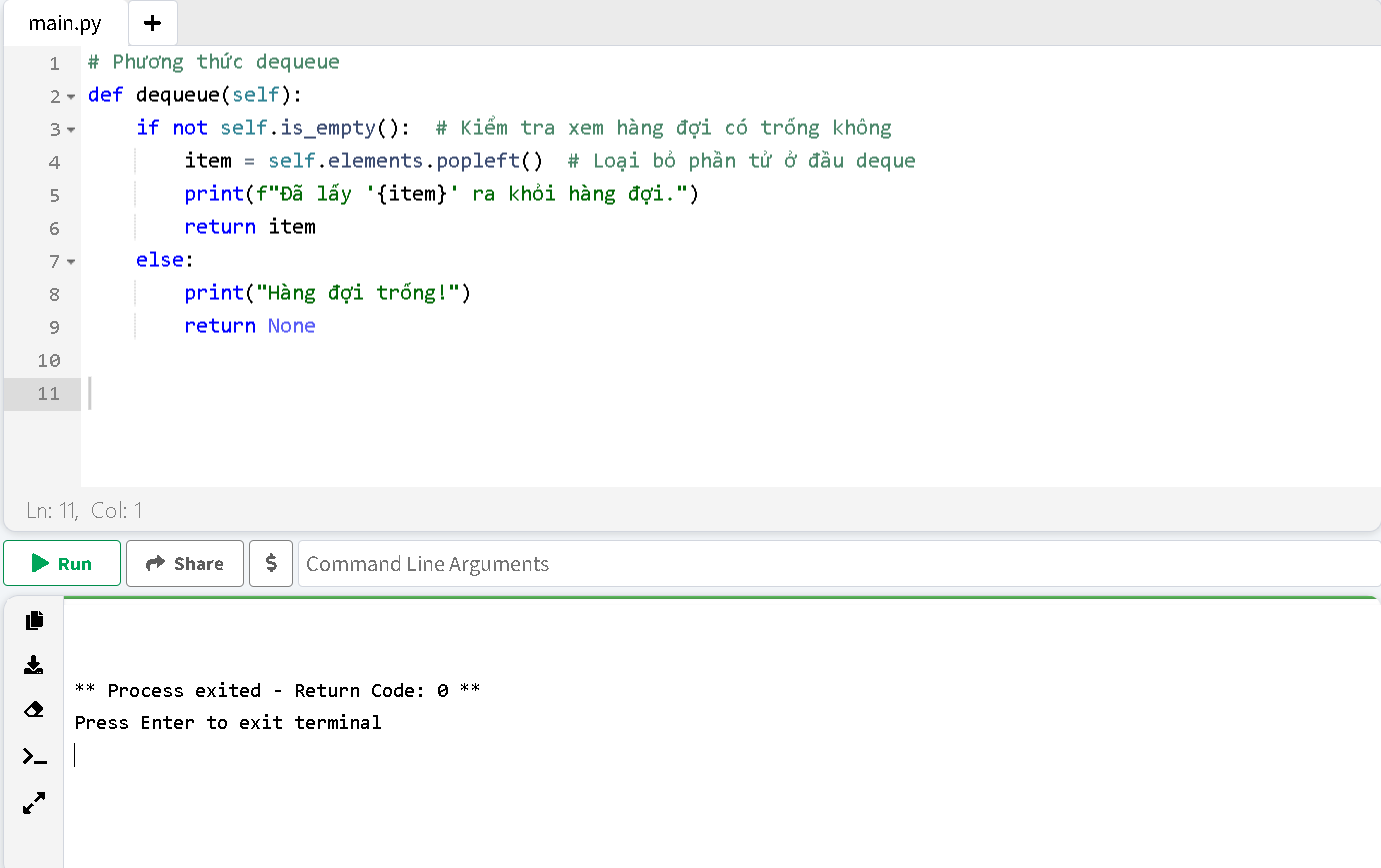


2.5.4

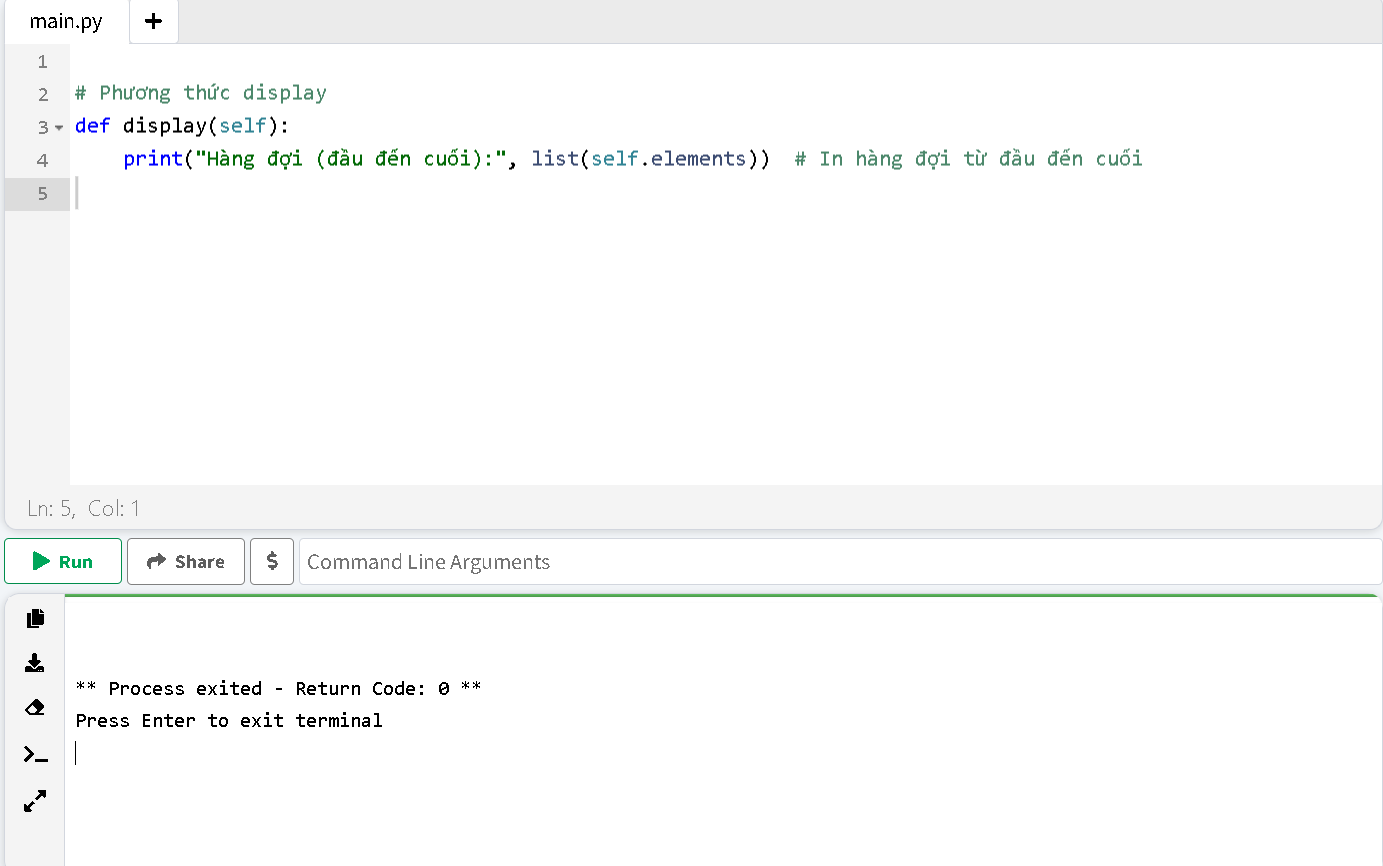


3.3.1(deque)

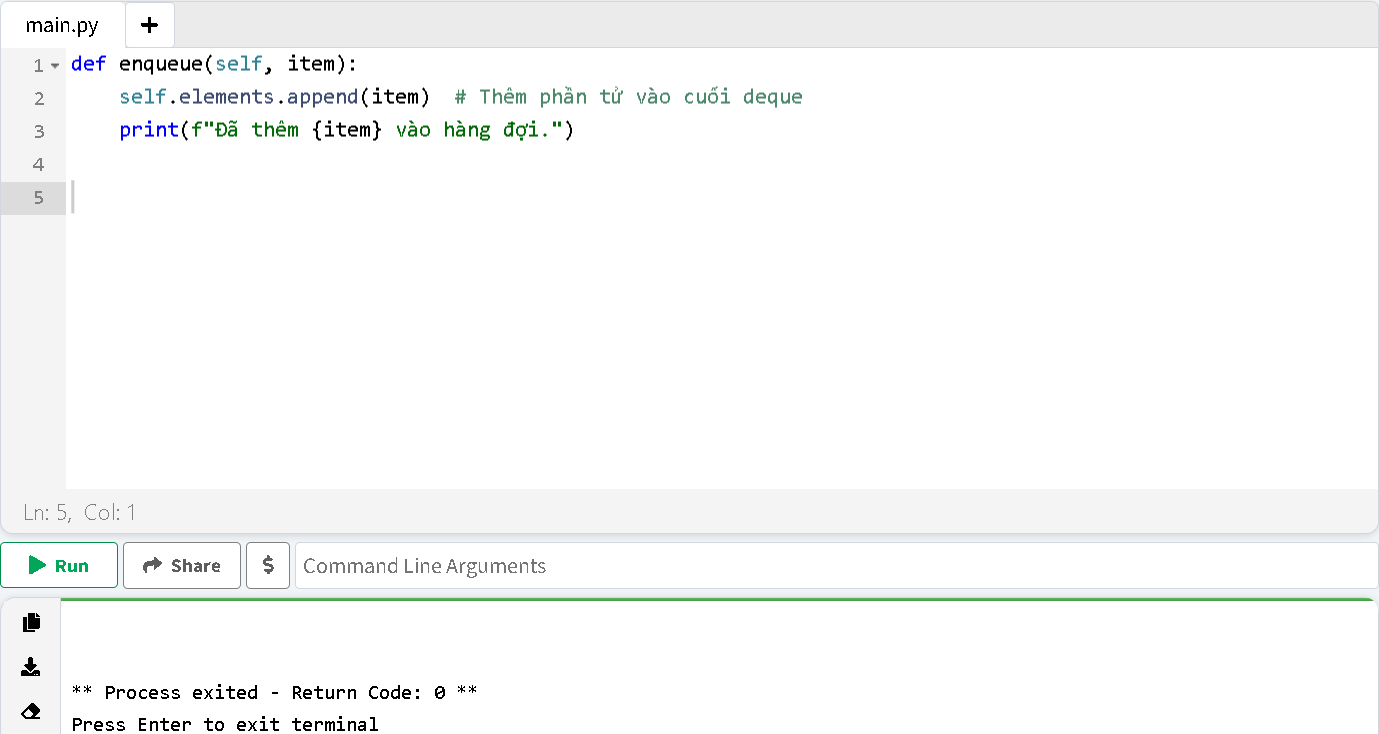
3.3.1(dequeue)



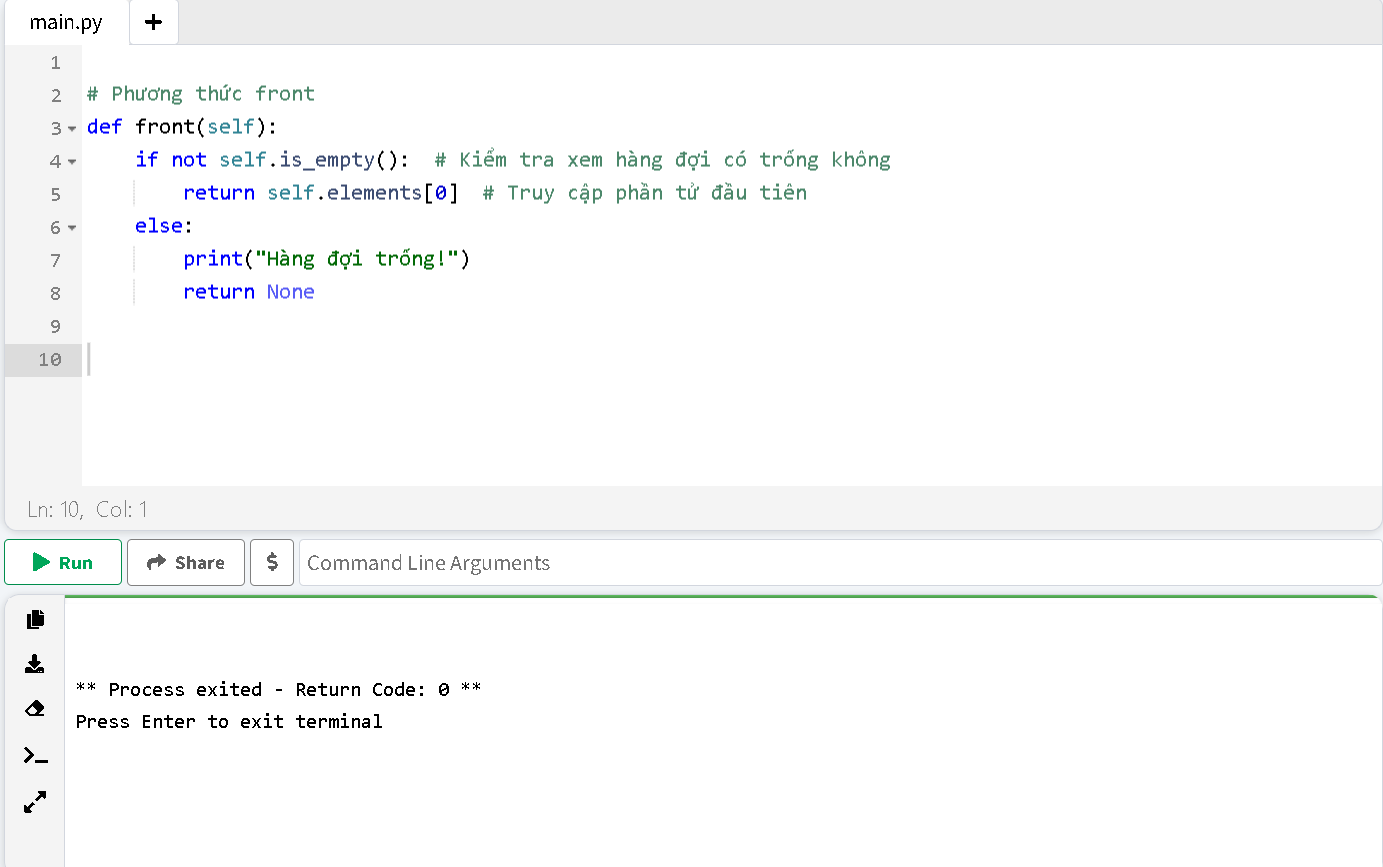
3.3.1(display)



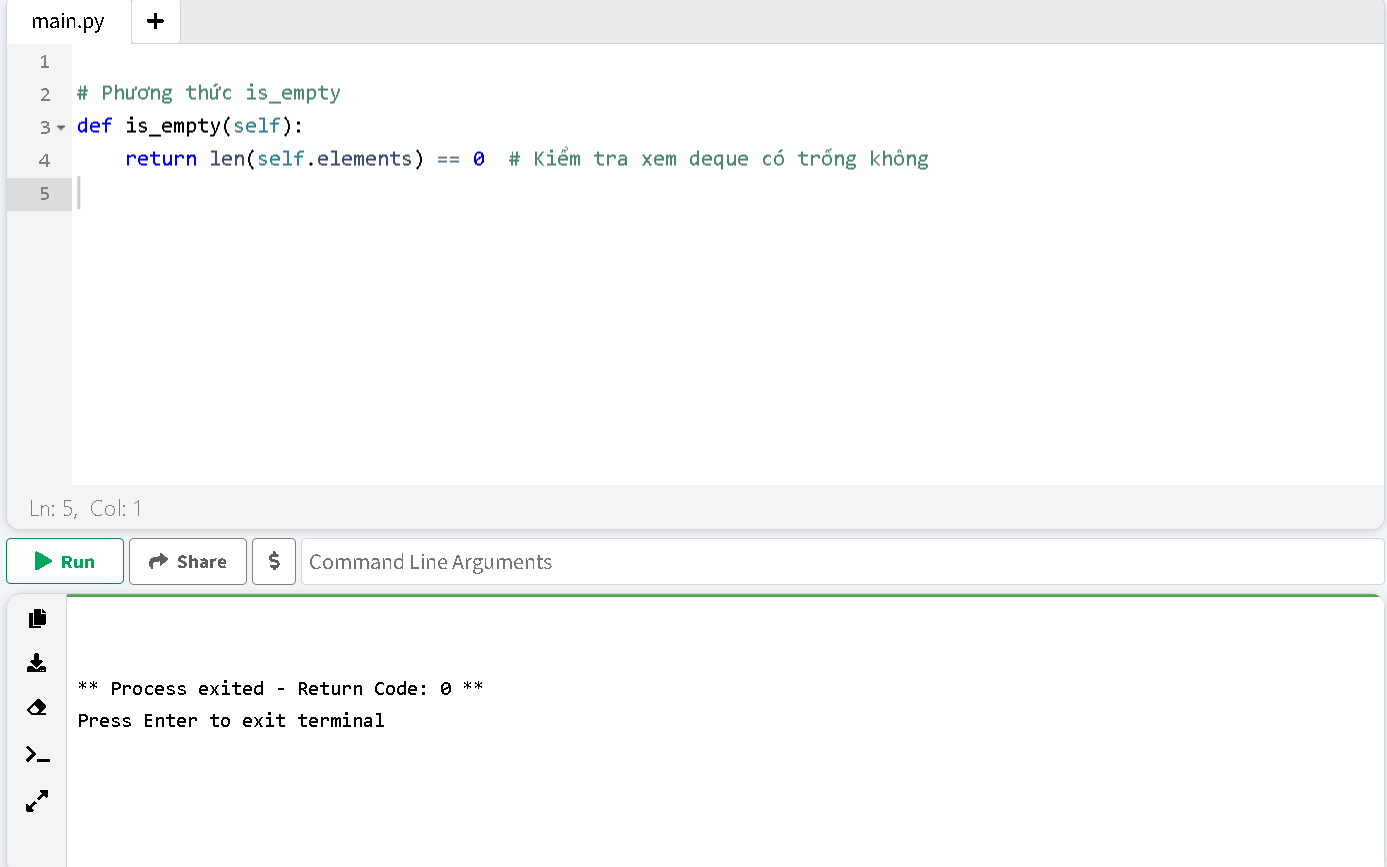
3.3.1(enqueu)



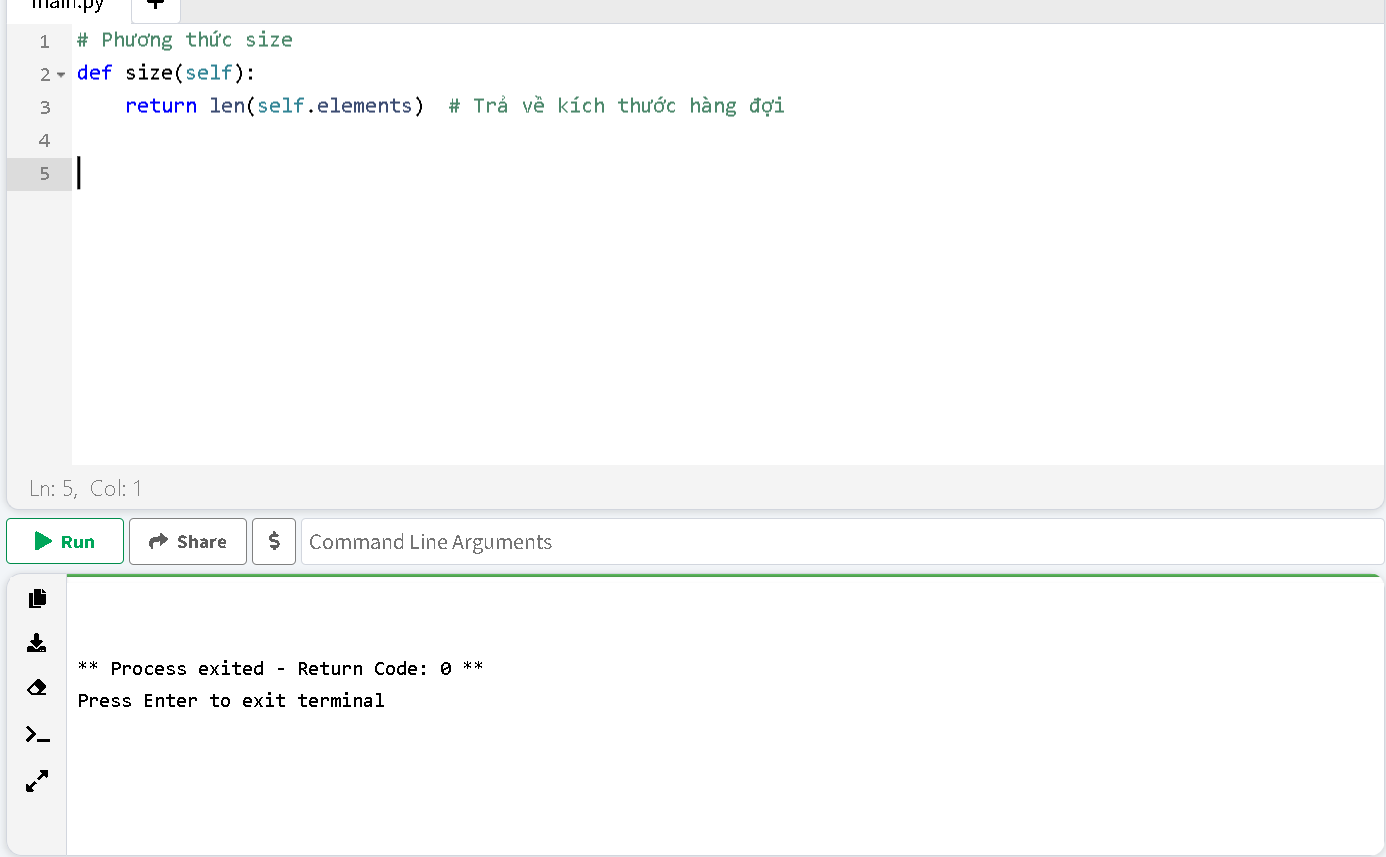
3.3.1(front)



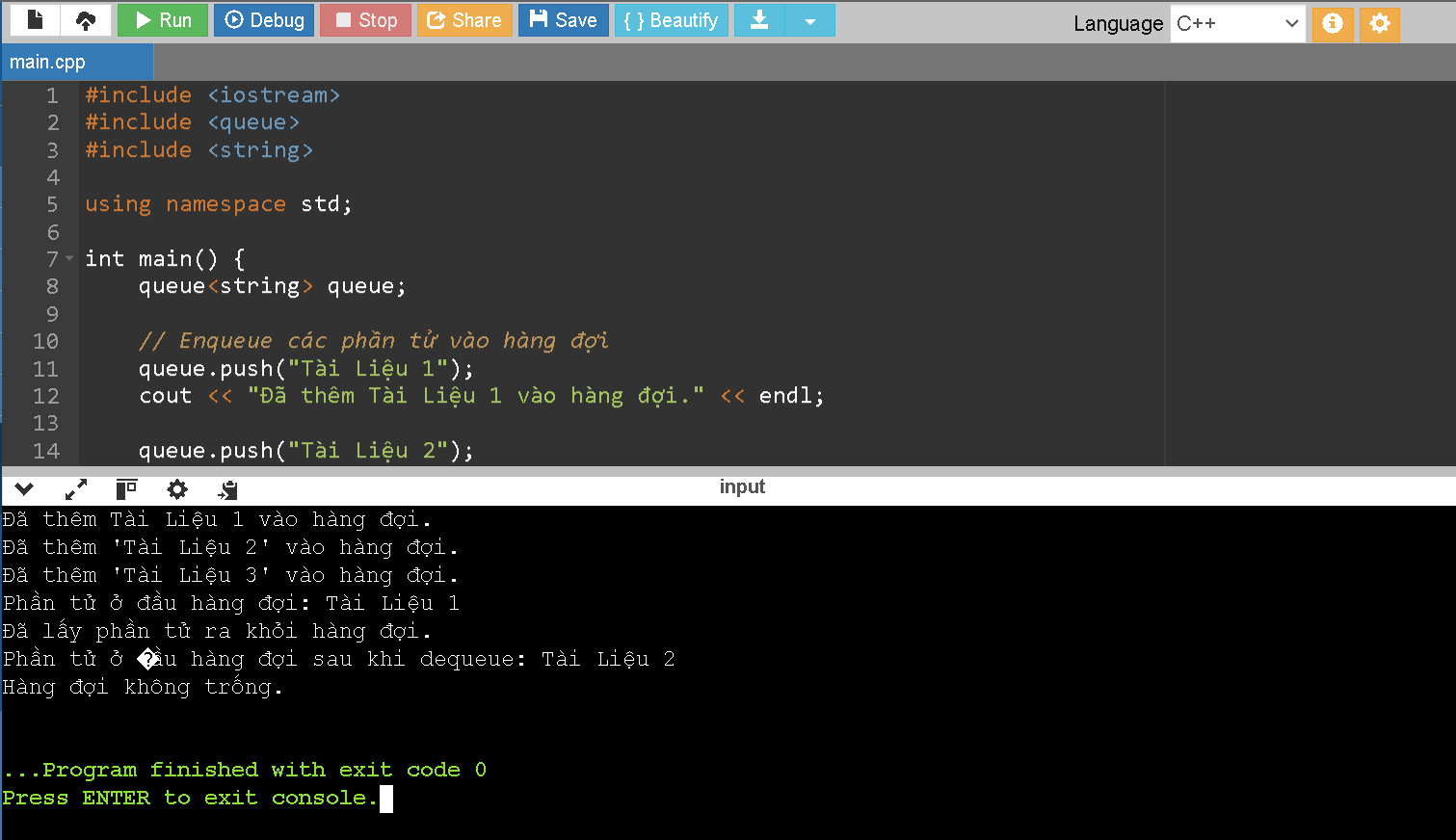
3.3.1(is\_empty)



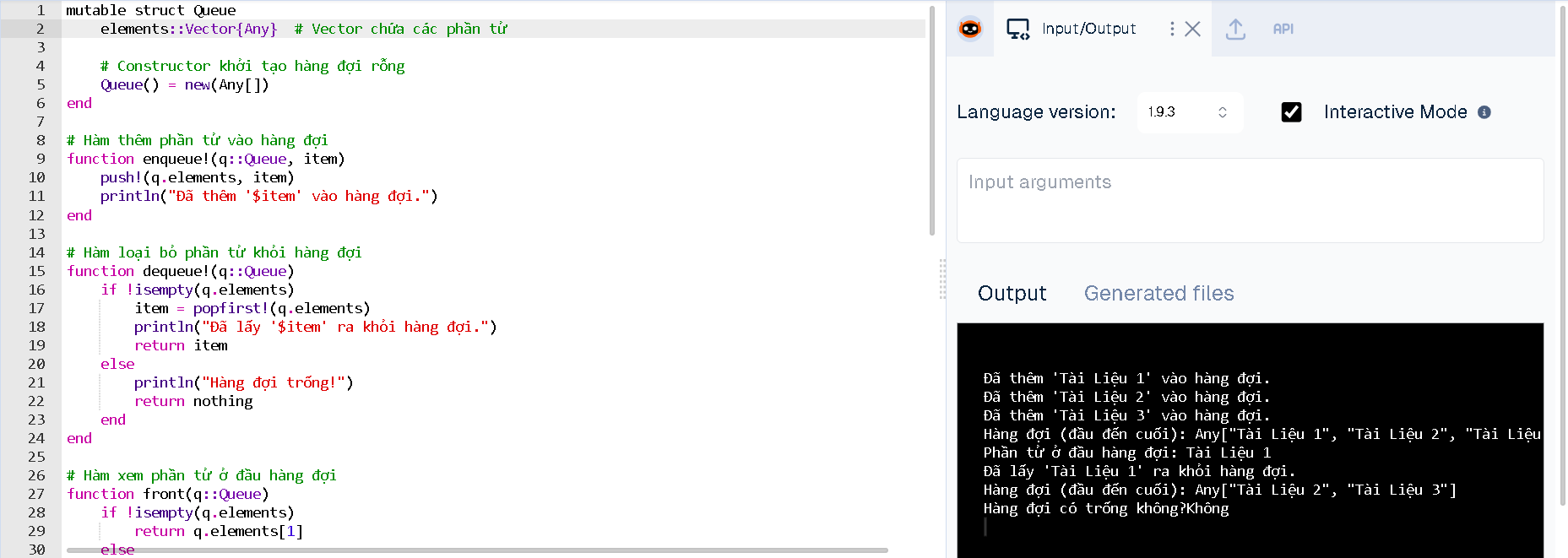
3.3.1(size)



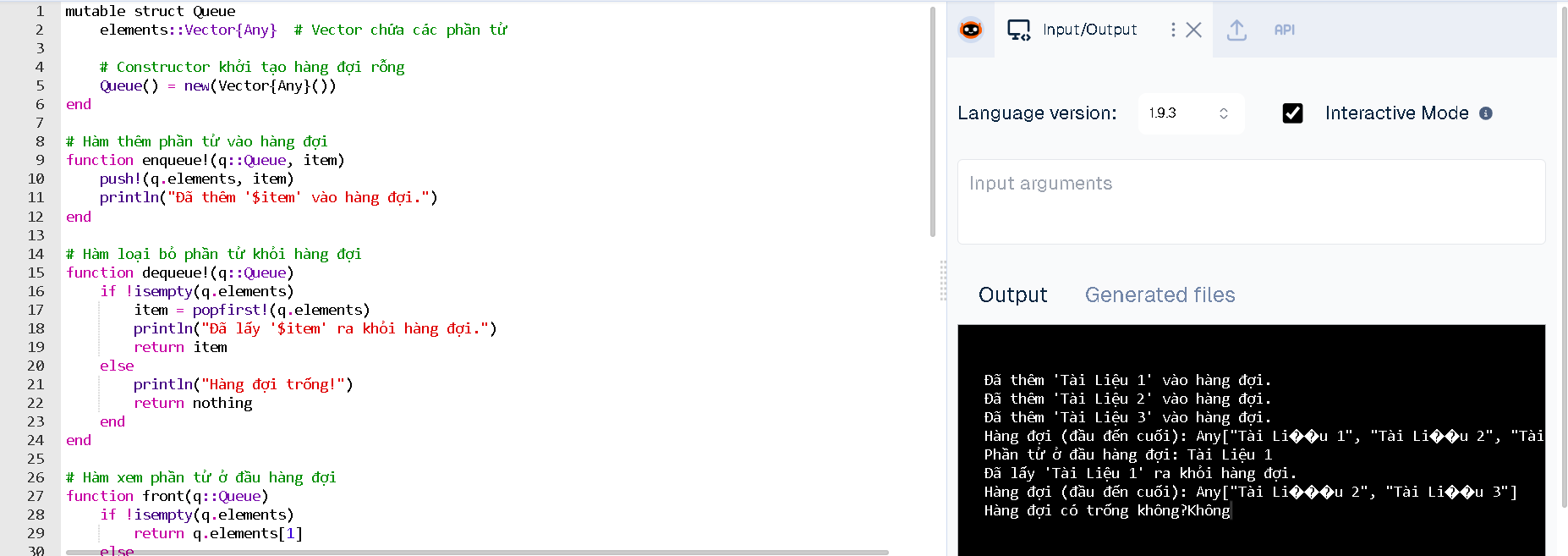
3.3.2



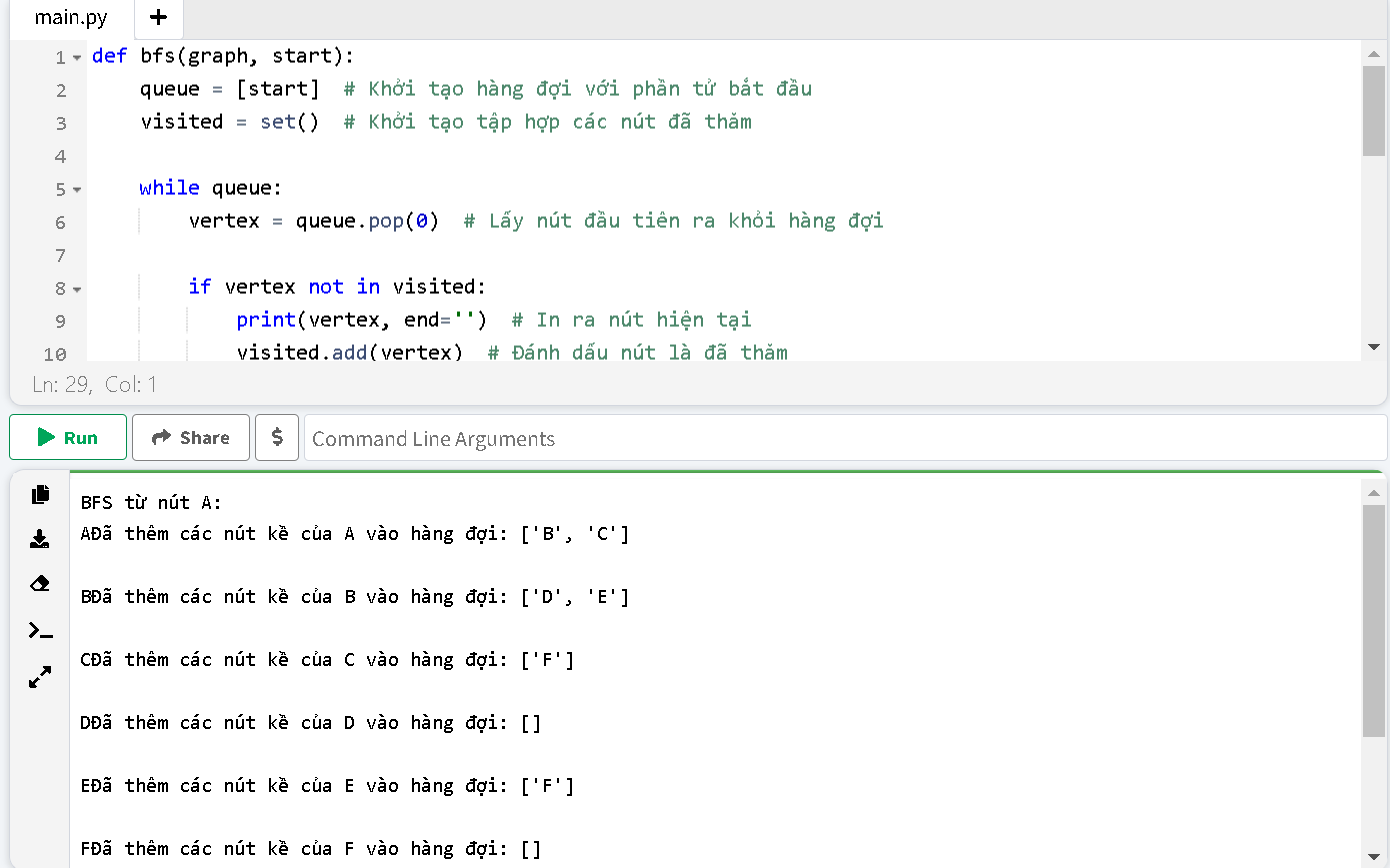
3.3.3.1



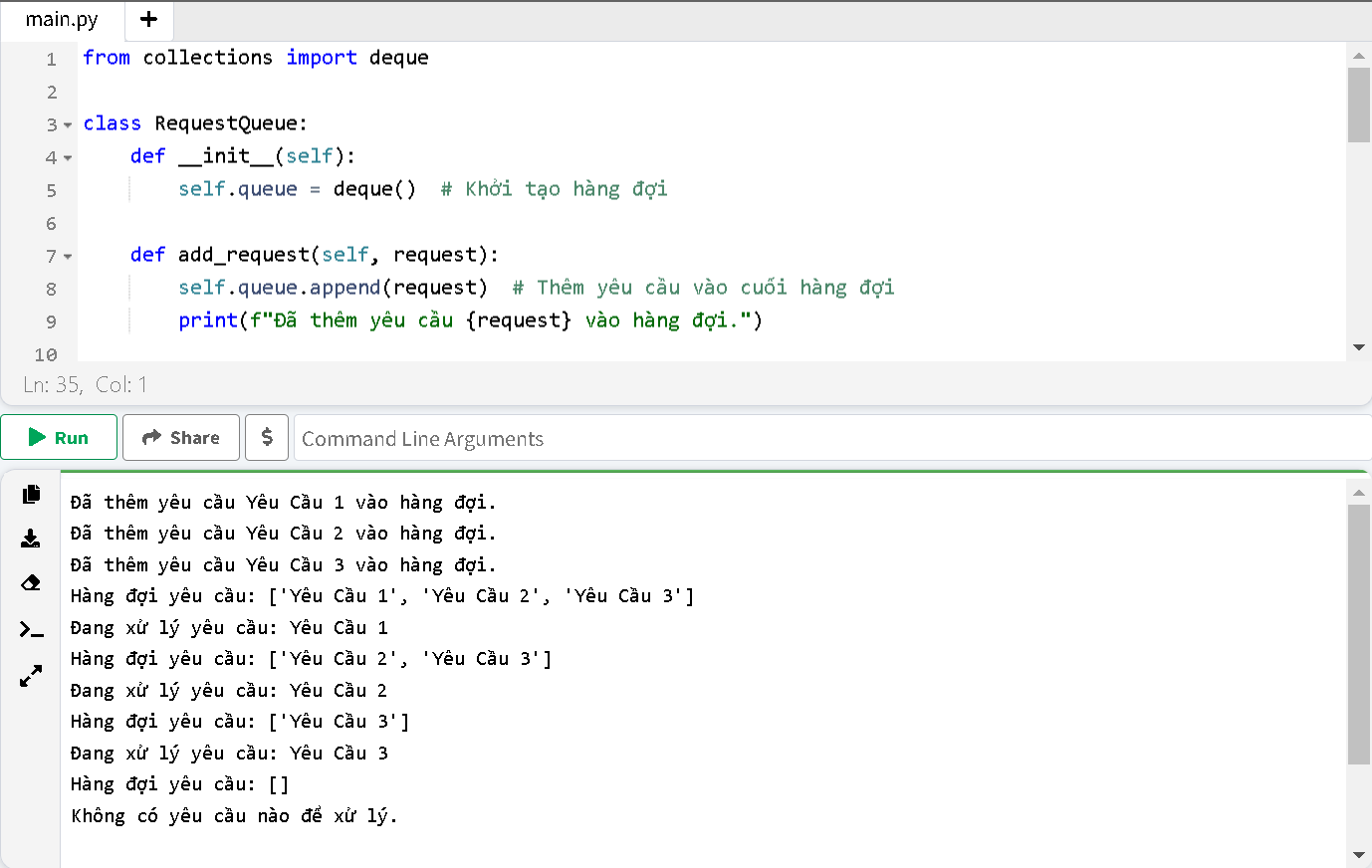
3.3.3.2



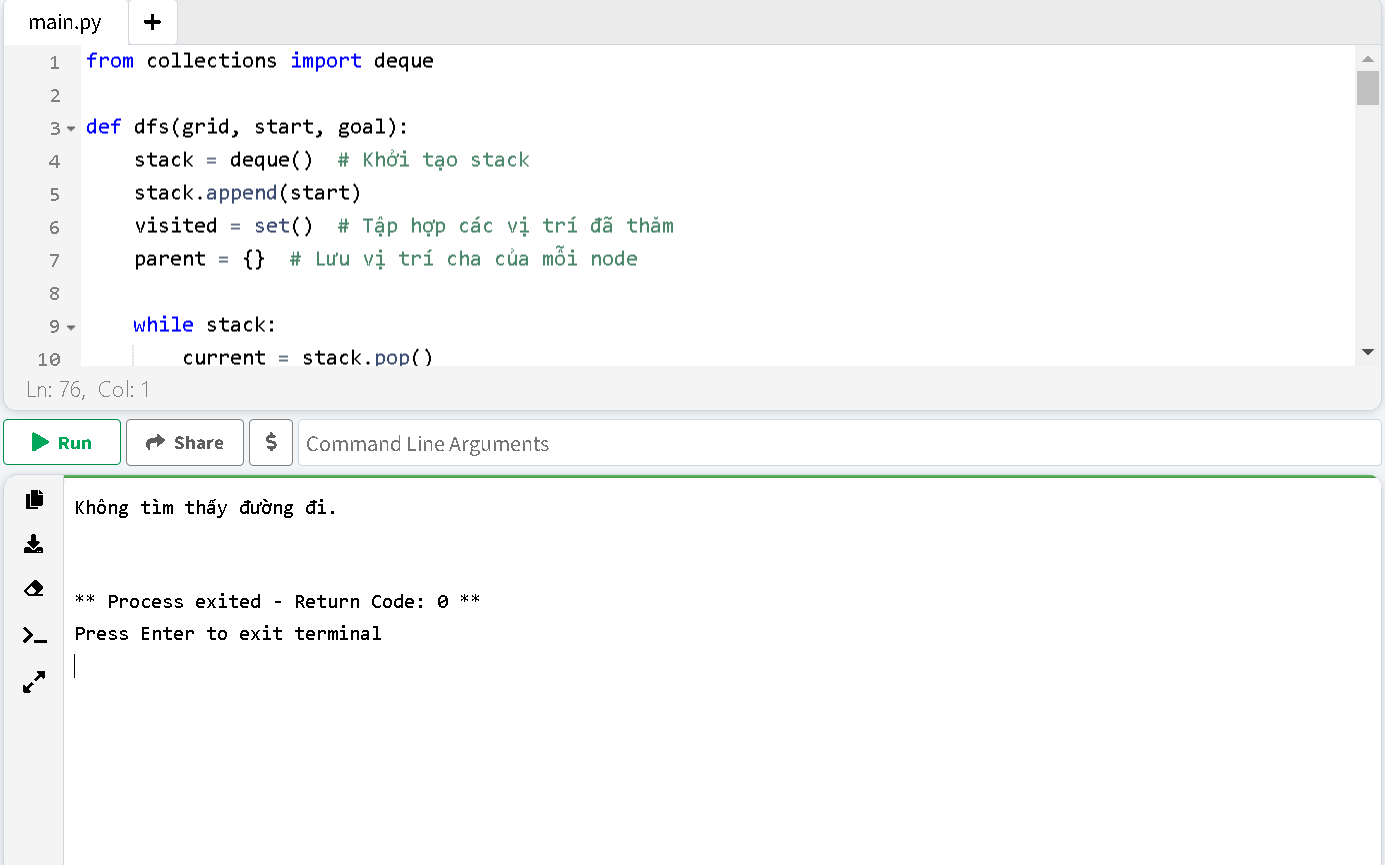
3.5.1



3.5.2

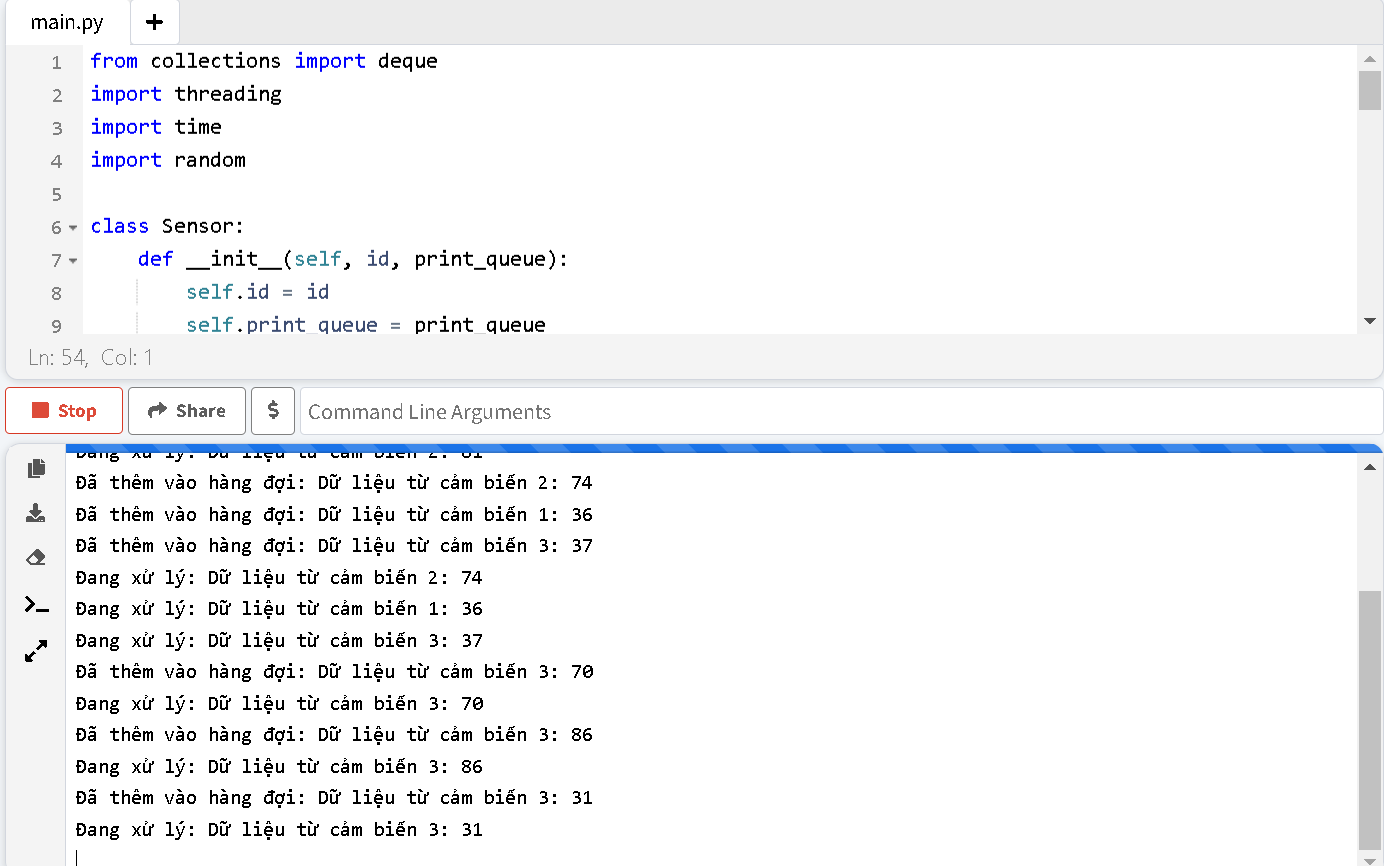


4.1



4.2

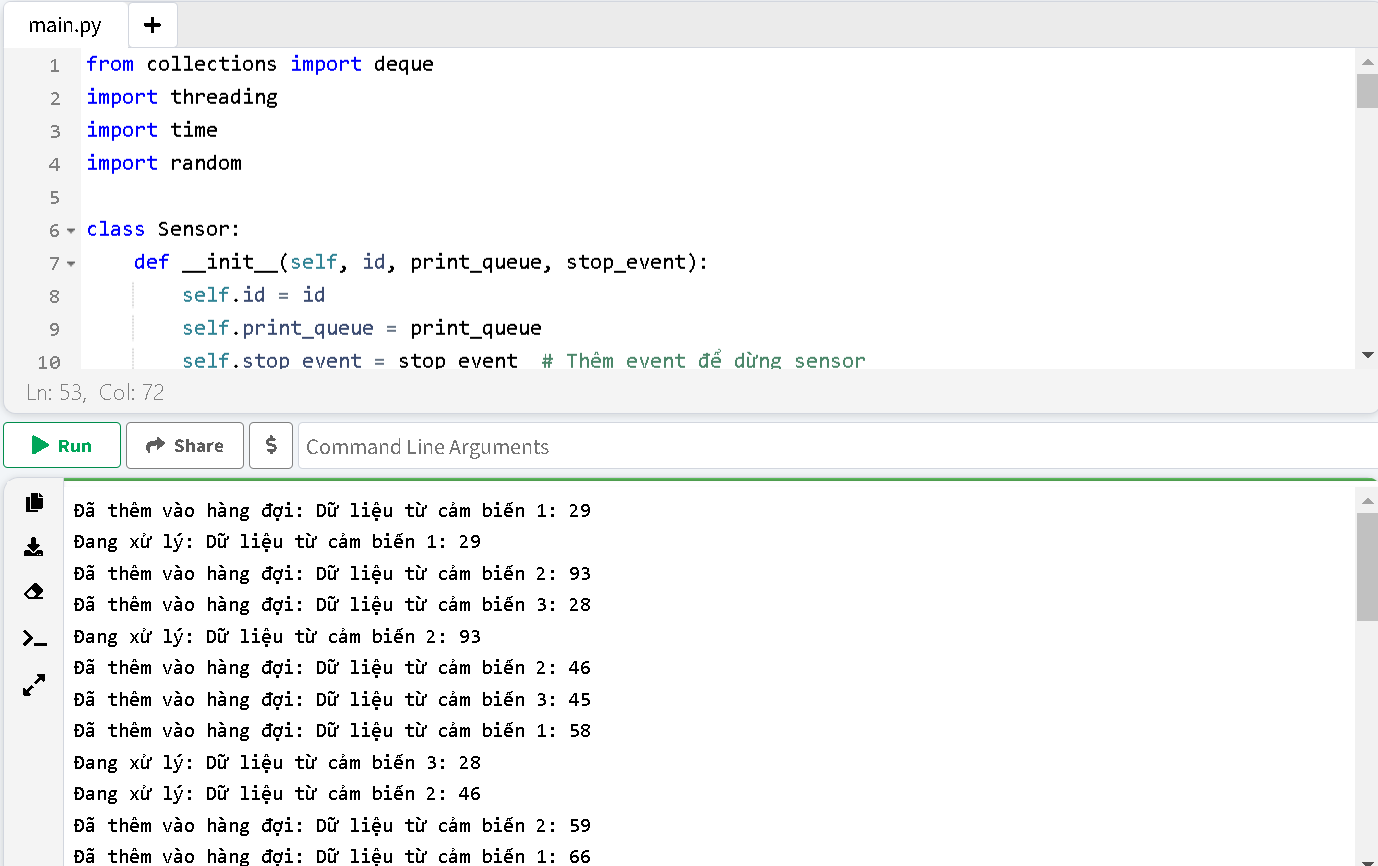
Bài tập gốc:

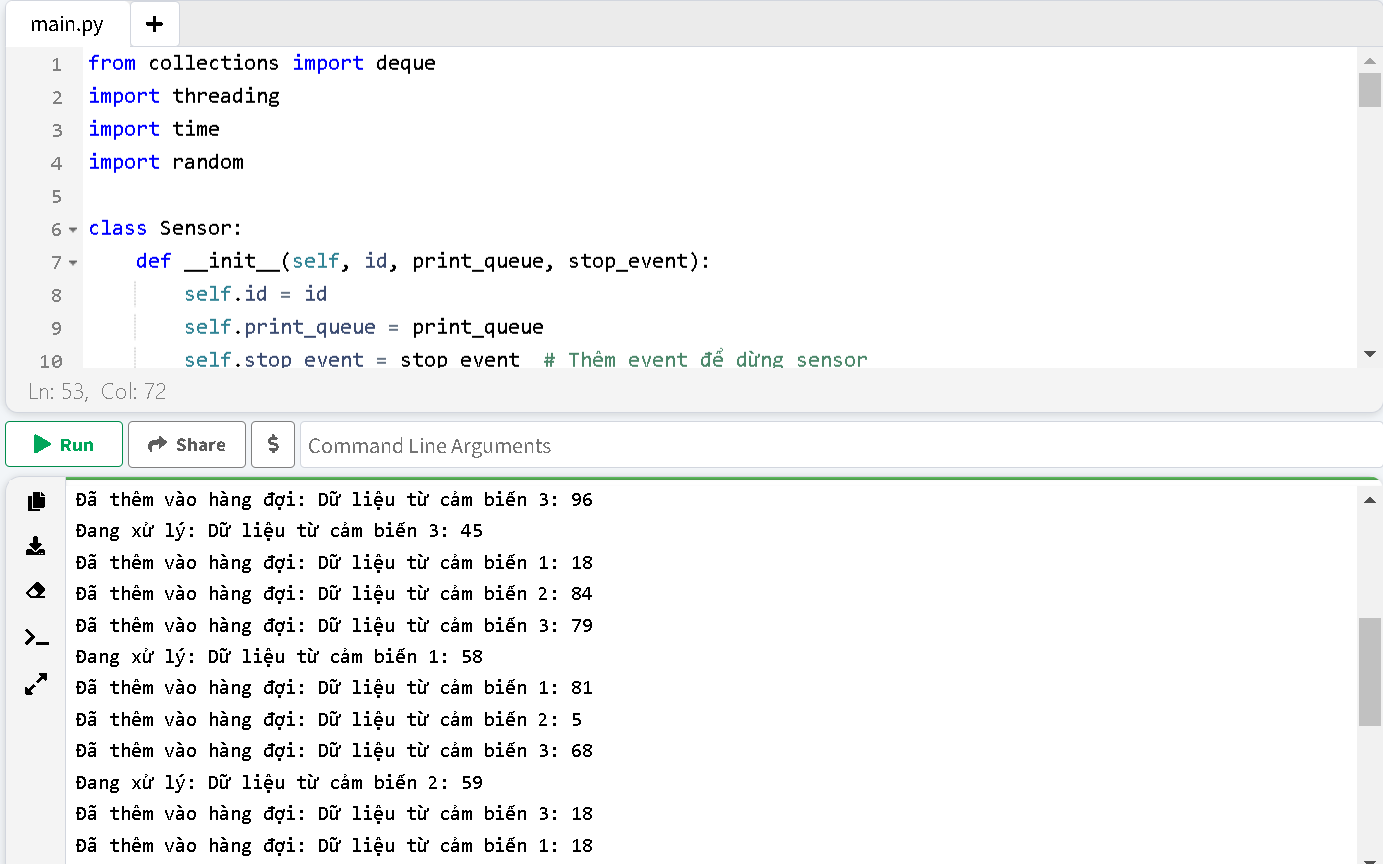


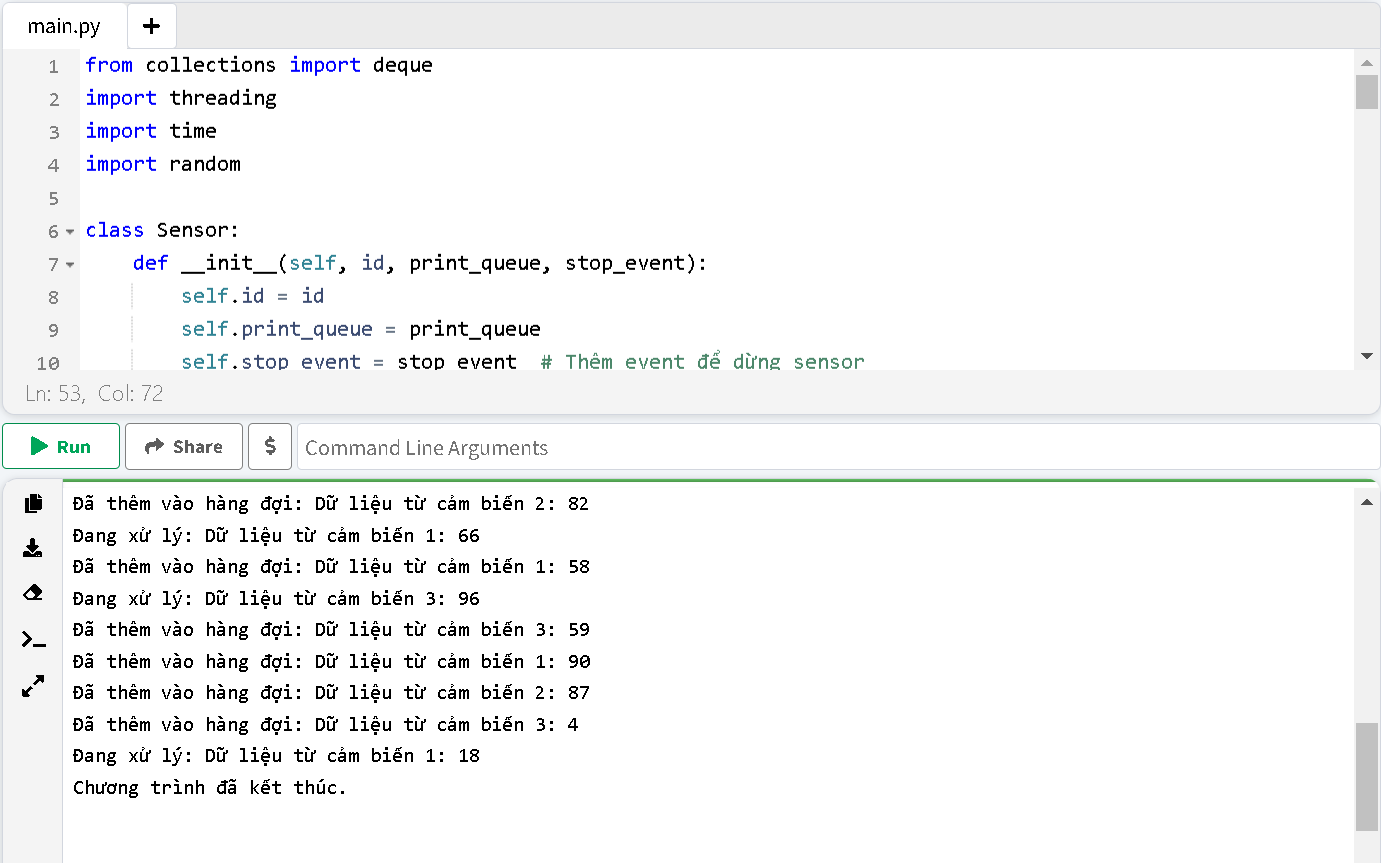
Kết quả: sau khi khởi chạy đoạn code thì sẽ xảy ra trường hợp là kết quả sẽ chạy khồng dừng lại.

Bài tập tự sửa:

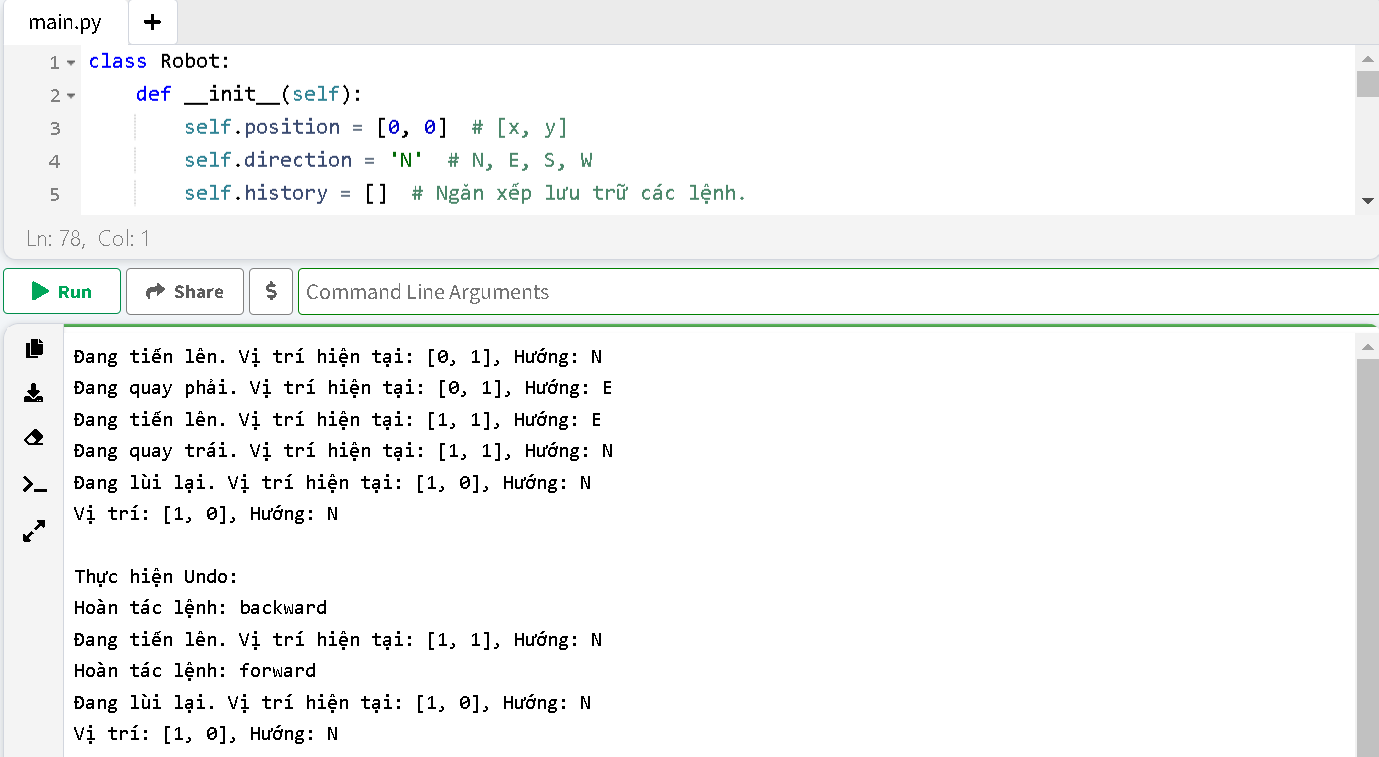
* Thêm threading.Condition để chờ khi hàng đợi trống – Tiết kiệm CPU thay vì chạy vòng lặp liên tục.
* Thêm threading.Event() để kiểm soát dừng chương trình – Có thể dừng chương trình sau 10 giây hoặc khi người dùng nhấn Ctrl + C.
* Các cảm biến chỉ chạy khi chương trình hoạt động – Khi stop\_event.set(), tất cả cảm biến và luồng xử lý sẽ dừng lại an toàn.



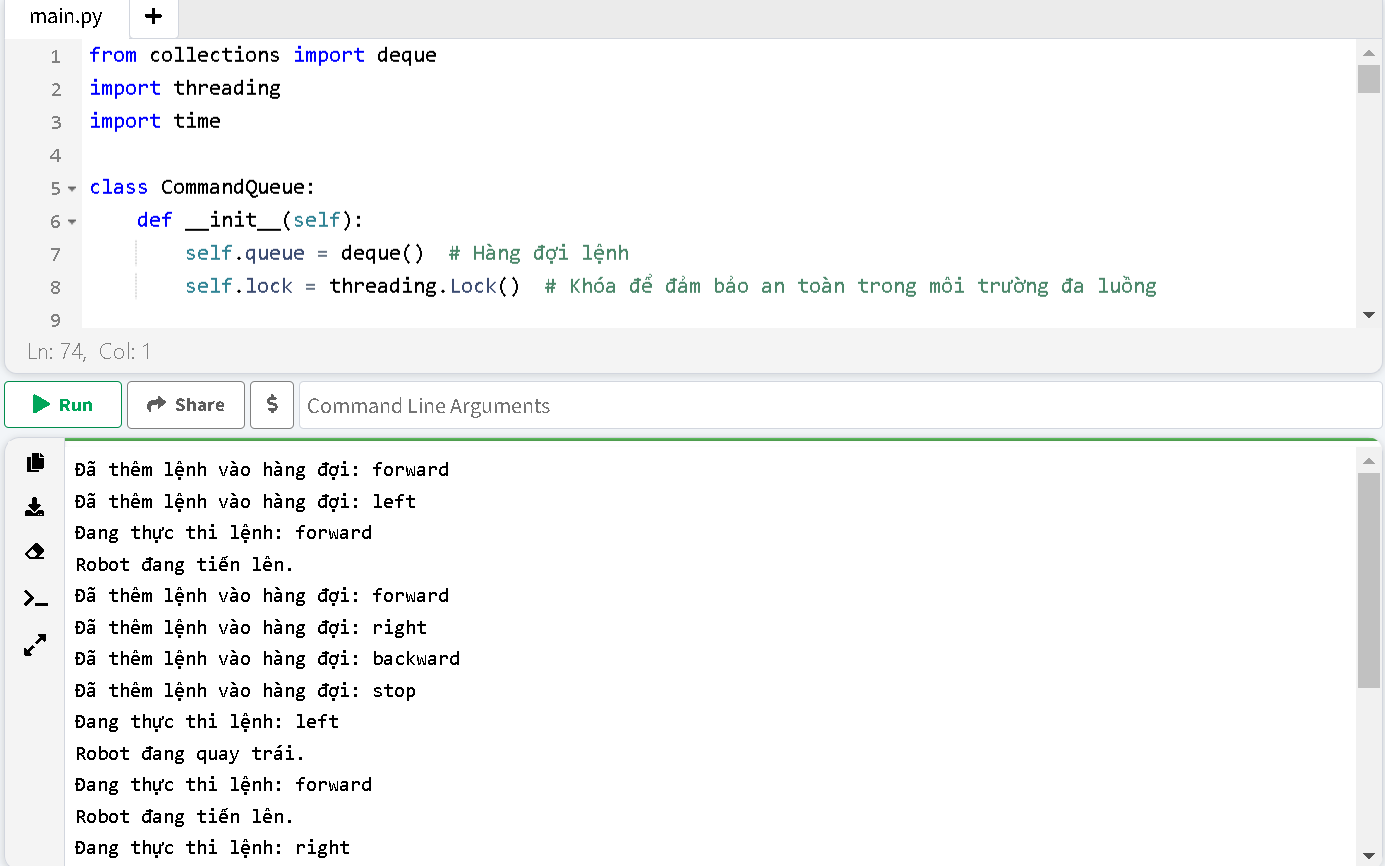


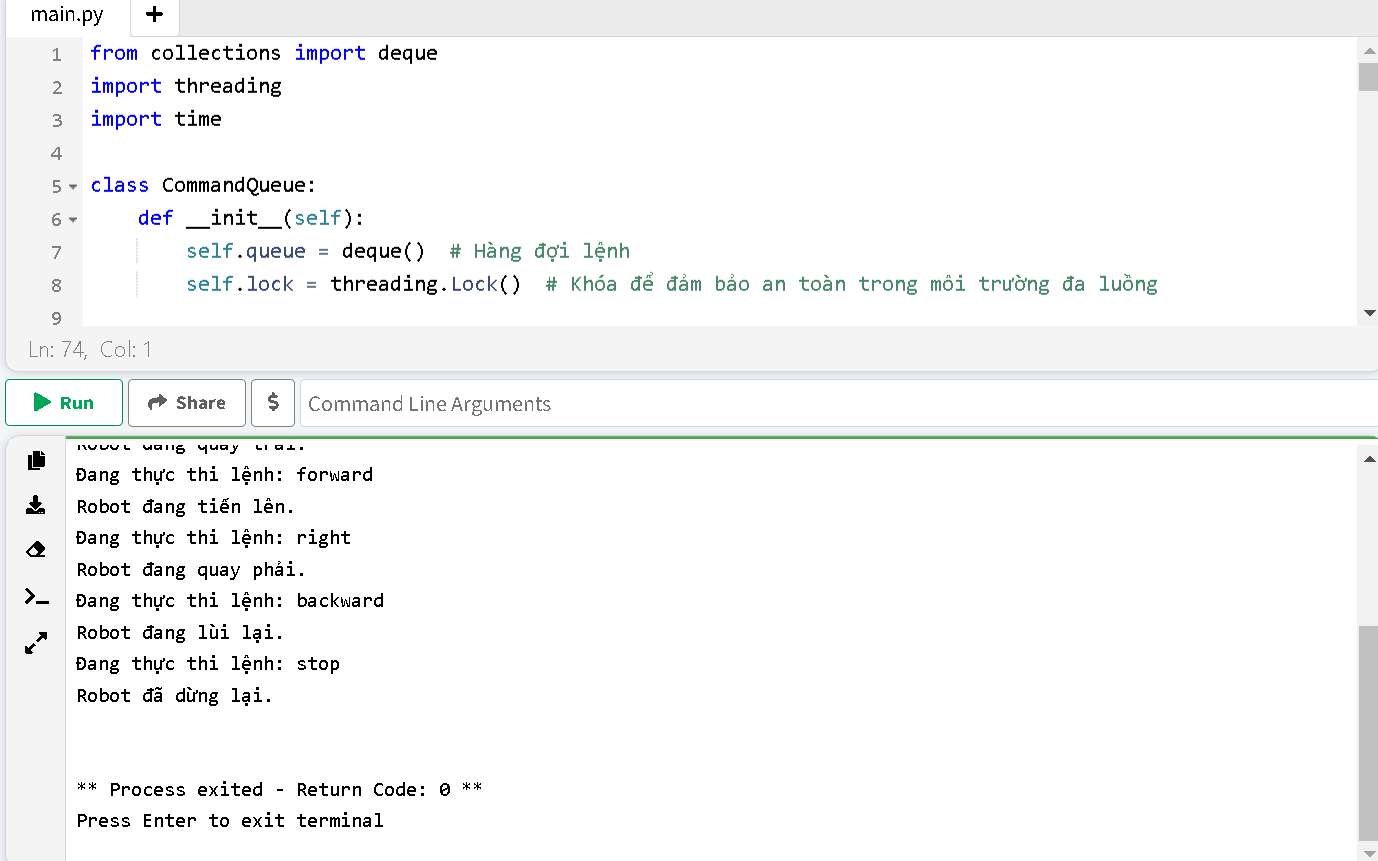


4.3

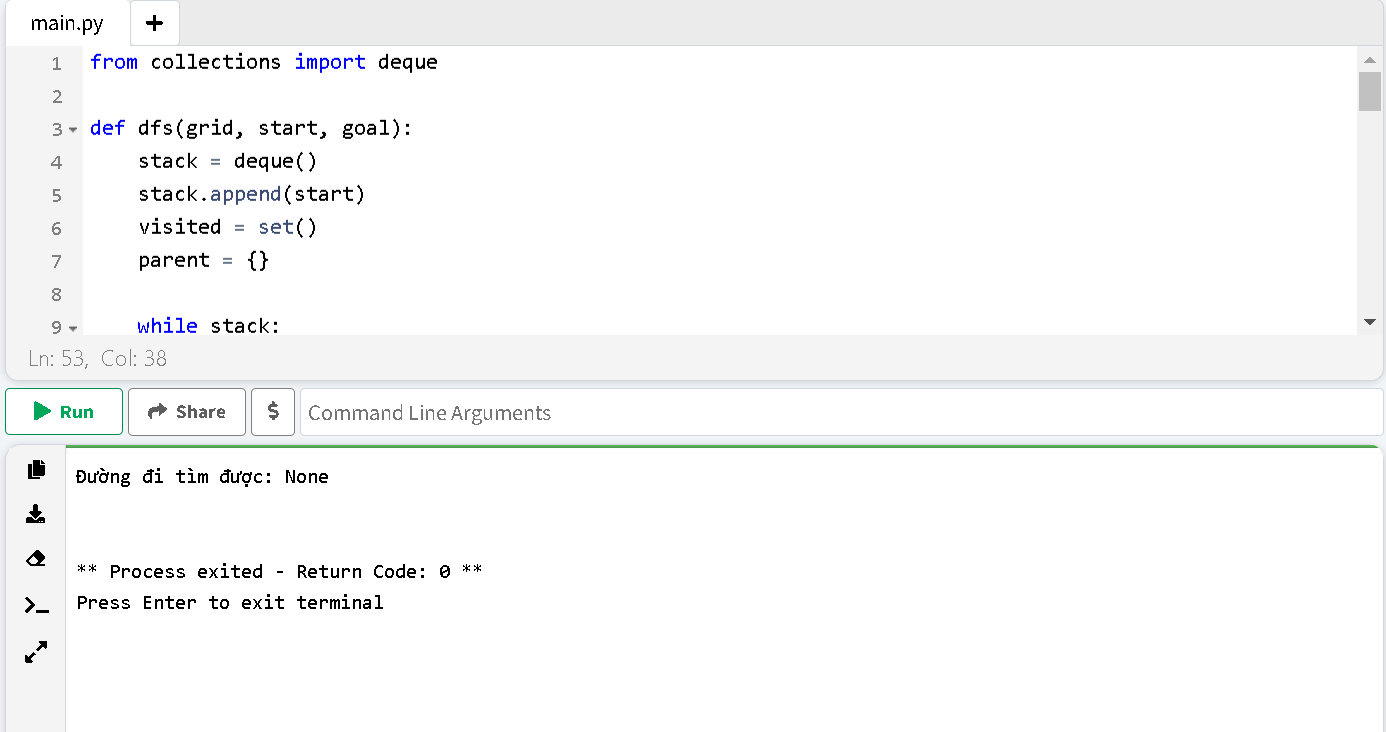


4.4

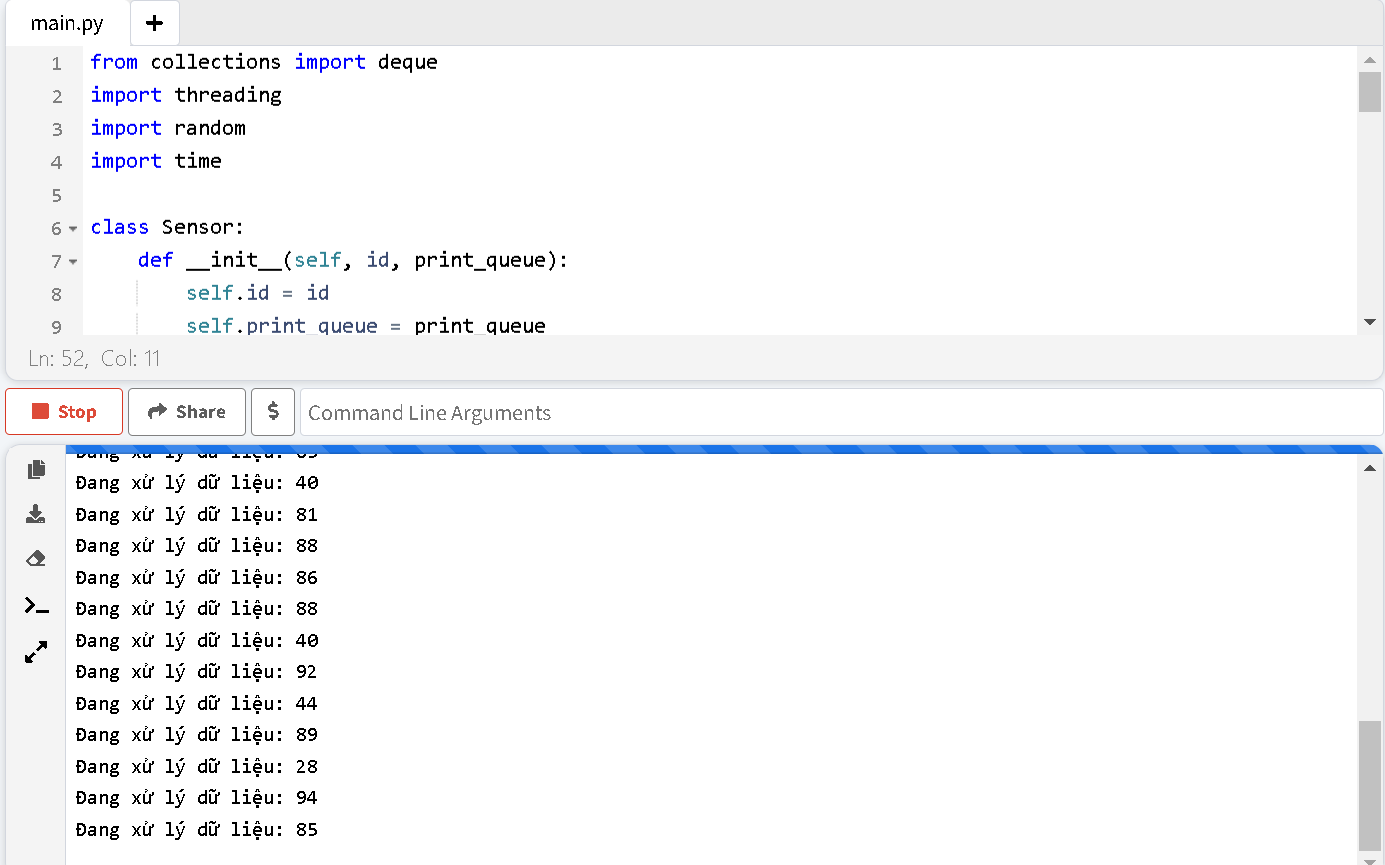




5.1



5.2



Kết quả: sau khi khởi chạy đoạn code thì nó sẽ chạy không ngừng

5.3



5.4

