**Ý tưởng**

Ký pháp nghịch đảo Ba Lan có 2 phần:

+ Chuyển biểu thức trung tố sang biểu thức hậu tố

+ Tính toán biểu thức hậu tố

**I. Chuyển biểu thức trung tố sang biểu thức hậu tố**

- Dùng stack để lưu trữ toán tử và kí tự ‘(’

- Dùng queue để chứa toán tử và toán hạng của biểu thức hậu tố

- Thiết lập thêm hàm orderOperator() để xác định thứ tự toán tử (độ ưu tiên toán tử):

* ‘^’ thì return 1.
* ‘\*’, ‘/’, ‘%’ thì return 2.
* ‘+’, ‘-’ thì return 3.
* Còn lại không phải toán tử thì return 4.

**Thuật toán:**

Chạy từ đầu chuỗi đến cuối chuỗi của biểu thức infix, ta xét từng kí tự infix[i]

1. Nếu là toán hạng thì push vào queue.
2. Nếu là ‘(’ thì push vào stack.
3. Nếu là ‘)’ thì lấy ra lần lượt phần tử ở đỉnh stack và push vào queue cho đến khi phần tử đỉnh stack là ‘(’ thì dừng lại (đây là điều kiện lặp). Sau đó loại bỏ ‘(’ ra khỏi stack.
4. Nếu là toán tử

* Nếu **stack không rỗng** và đồng thời đỉnh stack là toán tử có độ ưu tiên >= toán tử infix[i] thì lấy ra toán tử ở đỉnh stack push vào queue. Lặp lại cho đến khi điều kiện không còn thoả mãn.
* Sau đó push infix[i] vào trong stack.

Sau khi duyệt hết kí tự trong infix mà stack vẫn còn phần tử thì lấy ra hết lần lượt các phần tử ở đỉnh stack và push nó vào trong queue.

**II. Tính toán biểu thức hậu tố**

Dùng stack để lưu trữ toán hạng và kết quả sau khi tính toán 2 toán hạng

**Thuật toán:**

Lấy ra từng phần tử ở đầu queue và xét:

1. Nếu **không là toán tử** thì push nó vào stack
2. Nếu là toán tử thì lấy ra 2 toán hạng ở đỉnh stack. Thực hiện tính toán dựa theo toán tử, sau đó push kết quả vào stack.

**Lưu ý**: Khi tính toán ta lấy toán được lấy ra sau tính toán với toán hạng được lấy ra trước (lấy ra trước sau theo quy tắc LIFO của stack)

Tiếp tục thực hiện đến khi queue rỗng thì dừng

= > Sau khi duyệt hết phần tử của queue thì kết quả của biểu thức hậu tố là giá trị ở đỉnh stack (lúc này stack chỉ có 1 phần tử)