# Transform the Expression

## Đọc hiểu đề:

Bài toán yêu cầu chuyển các biểu thức đại số trung tố (infix) sang hậu tố (postfix).

Ta có yêu cầu đầu vào là :

t: số lượng biểu thức infix

ei -> en: ith biểu thức infix cụ thể

Ta có yêu cầu đầu ra là:

pi -> pn: ith biểu thức postfix tương ứng.

## Hướng giải quyết:

Ta sử dụng 1 biến p để lưu 1 biểu thức postfix.

Ta sử dụng 1 vector o để lưu các biểu thức postfix.

Ta sử dụng 1 stack e để lưu các toán tử và dấu mở ngoặc ‘(‘.

Có 1 điểm đáng lưu ý ở đây là ta nhận thấy đề bài đã giới hạn không tồn tại biểu thức 3 toán hạng trở lên trong 1 cặp ngoặc, nên sẽ không bao giờ tồn tại 2 toán tử liền kề được lưu stack. Từ đó việc kiểm tra độ ưu tiên cho toán tử trở nên dễ dàng hơn, chỉ cần so sánh với phần tử đầu tiên của stack là được.

Ta sẽ kiểm tra lần lượt từng ký tự s trong mỗi biểu thức infix.

* Nếu s là 1 toán hạng => thêm s vào p.
* Nếu s là dấu mở ngoặc ‘(‘ lưu s vào stack e.
* Nếu s là dấu đóng ngoặc ‘)’, lần lượt lấy các phần tử ra khỏi stack tới khi gặp ‘(‘, lấy cả ‘(‘ ra khỏi stack. Lần lượt thêm các toán tử đã lấy ra vào p.
* Nếu s là 1 toán tử với s < e.front(), lấy e.front() ra khỏi stack, thêm vào p.
* Nếu s là 1 toán tử với s > e.front(), thêm s vào p.
* Đẩy p vào o.

Lăp lại quy trình cho tới khi toàn bộ t số lượng biểu thức infix được kiểm tra.