

Đừng bỏ cuộc

Hiện tại, bài tập này đã có trên online judge chính thức của VNOI, bạn có thể truy cập ở đây: <https://oj.vnoi.info/problem/vmqpofi>

Những chuyện xảy ra trước đó...

Công ty Cốc Cốc (nhà tài trợ của cuộc thi) đang tập trung nghiên cứu trong lĩnh vực xử lý ngôn ngữ Tiếng Việt.

Pirate cũng đang rất quan tâm vấn đề này vì gần đây anh thường bị chóng mặt và nôn mửa khi phải xử lý một khối lượng bài tập TC (không phải TopCoder mà là TeenCode) quá lớn. Vì TeenCode quá rối rắm nên Pirate dễ thương và tội nghiệp quyết định tìm một vài phút giây thanh thản bên những biểu thức số học.

Nhắc đến biểu thức số học, chắc các bạn đã khá quen thuộc với khái niệm **Postfix Notation** (PN) (cũng được biết đến với cái tên **Reverse Polish Notation** - ký pháp nghịch đảo Ba Lan).

Trong ký pháp thông thường, toán tử được viết ở giữa hai toán hạng, ví dụ $3 + 4$. Tuy nhiên, trong PN, toán tử được viết sau hai toán hạng, ví dụ $3 4 +$. Các bạn sẽ tự hỏi rằng tại sao cách viết như vậy hiệu quả hơn? Một trong số các câu trả lời là vì biểu thức dạng PN có thể dễ dàng được tính toán bằng cấu trúc dữ liệu [ngăn xếp \(stack\)](#) với độ phức tạp tuyến tính.

Thuật toán tính giá trị của biểu thức dạng PN sử dụng stack như sau:

Xử lý biểu thức từ trái sang phải:

- Nếu gặp một toán hạng, đẩy (push) nó vào stack.
- Nếu gặp một toán tử, lấy (pop) hai toán hạng từ stack ra và tính kết quả của phép tính gồm hai toán hạng trên và toán tử đang xét. Sau đó, đẩy (push) kết quả trở lại vào stack.

Hãy xem xét một ví dụ phức tạp hơn: " $4 - 3 * 2$ ". PN của biểu thức trên là " $4 3 2 * -$ ". Ta sẽ thử áp dụng thuật toán vào ví dụ:

Xử lý biểu thức từ trái sang phải:

- Đẩy 4 vào stack.

- Đẩy 3 vào stack.

- Đẩy 2 vào stack.

- Gặp dấu $*$, ta lấy 3 và 2 ra khỏi stack, tính $3 * 2 = 6$. Đẩy 6 vào stack.

- Gặp dấu $-$, lấy 4 và 6 ra, tính $4 - 6 = -2$ (lưu ý rằng số nào được đưa vào stack trước sẽ đứng trước trong phép tính). Đẩy -2 vào stack.

- Kết quả của biểu thức là -2.

Pirate loay hoay tìm cách mô phỏng thuật toán trên. Anh tìm thấy một cái hộp đựng cầu lông. Tốt quá rồi, cái này có thể mô phỏng được stack. Nhưng khổ một nỗi là hộp đựng cầu này không có cái nắp nào cả nên hai đầu trống trơn. Kế hoạch thất bại, nếu phải quay lại với TeenCode thì chết còn hơn, Pirate nghĩ thầm. Trong lúc đang dùng hộp cầu đập vào đầu để tự vẫn, Thần Cầu hiện ra cười và hỏi:

- *Taj seo kon hú? (Thần Cầu cũng thích TeenCode)*

- *Hú hú hú, con không thể dùng cái hộp này để làm một cái stack vì nó không có đáy.*

- *Đừng bỏ cuộc con ạ, người nhìn thấy thành công là người nhìn thấy khó khăn và biến nó thành cơ hội. Con hãy nhìn lại xem cái hộp giờ trông giống cái gì?*

- *Nó giống... một hàng đợi ạ.*

Thần Cầu đã cho Pirate một gợi ý tuyệt vời. Thay vì làm thí nghiệm với stack, Pirate sẽ làm thí nghiệm với [queue \(hàng đợi\)](#). Nhưng như thế thì Pirate cần một ký pháp mới để biểu diễn biểu thức. Anh gọi ký pháp mới là **Queue Postfix Notation (QPN)**. Thuật toán tính giá trị của QPN cũng gần giống với PN thông thường:

Xử lý biểu thức từ trái sang phải:

- *Nếu gặp một toán hạng, đẩy (enqueue) nó vào queue.*
- *Nếu gặp một toán tử, lấy (dequeue) hai toán hạng từ queue ra và tính kết quả của phép tính gồm hai toán hạng trên và toán tử đang xét. Sau đó, đẩy (enqueue) kết quả trở lại vào queue.*

Biểu thức " $4 - 3 * 2$ " có thể được viết lại dưới dạng QPN là " $3 2 4 * -$ " và được tính toán như sau:

Xử lý biểu thức từ trái sang phải:

- *Đẩy 3 vào queue.*

- *Đẩy 2 vào queue.*

- *Đẩy 4 vào queue.*

- *Gặp *, lấy 3 và 2 ra khỏi queue, tính $3 * 2 = 6$. Đẩy 6 vào queue.*

- *Gặp -, lấy 4 và 6 ra khỏi queue, tính $4 - 6 = -2$ (lưu ý rằng số nào vào queue trước sẽ đứng trước trong phép tính). Đẩy -2 vào queue.*

- *Kết quả của biểu thức là -2.*

Và bây giờ...

Bạn sẽ giúp Pirate thực hiện thí nghiệm của anh ấy. Bạn được giao cho một biểu thức đúng dạng PN. Biểu thức gồm một số các toán hạng và các toán tử cộng, trừ, nhân.

- **Thử thách thứ nhất** của bạn là **sử dụng tất cả toán hạng và toán tử** của biểu thức đã cho để sắp xếp thành một biểu thức mới dưới dạng QPN. Khi sử dụng thuật toán QPN để tính toán biểu thức mới sẽ thu được **kết quả tương tự** như khi tính toán biểu thức đã cho

bằng thuật toán PN.

- **Thử thách thứ hai** của bạn cũng là tạo ra một biểu thức QPN có kết quả tương tự với biểu thức PN được cho. Bạn vẫn phải **sử dụng tất cả các toán hạng** trong biểu thức ban đầu. Điều đặc biệt là bạn có quyền **đổi dấu các toán hạng** (số âm thành số dương và ngược lại). Ngoài ra, bạn có quyền **sử dụng toán tử một cách tự do nhưng không được dùng dấu trừ**. Nói cách khác, bạn chỉ được sử dụng dấu cộng và dấu nhân.

Bạn có thể chọn giải một trong hai thử thách hoặc cả hai. Chúc bạn may mắn!

Input

Input gồm hai dòng:

- Dòng thứ nhất chứa số **N**, độ dài của biểu thức.
- Dòng thứ hai gồm **N** phần tử: mỗi phần tử là một số nguyên hoặc một trong các dấu cộng (+), trừ (-) hoặc nhân (*), thể hiện một biểu thức dưới dạng Postfix Notation.

Output

Output gồm hai dòng:

- Dòng thứ nhất gồm **N** phần tử: một biểu thức Queue Postfix Notation là câu trả lời cho thử thách thứ nhất.
- Dòng thứ hai gồm **N** phần tử: một biểu thức Queue Postfix Notation là câu trả lời cho thử thách thứ hai.

Nếu bạn không trả lời thử thách nào, bạn phải in ra N số 0 ở dòng tương ứng.

Example

Input:

```
5  
4 3 2 * -
```

Output:

```
3 2 4 * -  
-3 2 4 * +
```

Lưu ý: vì kết quả của biểu thức có thể rất lớn, trong chương trình chấm bài, kết quả của mỗi biểu thức sẽ được lấy modulo $10^9 + 7$ để so sánh với nhau.

Giới hạn

- $N \leq 100001$. Trong 30% số test, $N \leq 9$.
- Biểu thức ở Input được đảm bảo là một biểu thức dạng Postfix Notation đúng (luôn có kết quả).
- Mỗi toán hạng trong biểu thức ở Input là một số nguyên có trị tuyệt đối không vượt quá 10^4 .

Cách tính điểm

- Nếu chỉ trả lời đúng một trong hai thử thách, bạn được 80% số điểm của test.
- Nếu trả lời đúng cả hai thử thách, bạn được 100% số điểm của test.
- Nếu bạn không in N số 0 ở dòng tương ứng với thử thách nào, tức là bạn chọn trả lời thử thách đó. Trong trường hợp này, nếu trả lời sai thử thách đó thì bạn sẽ được 0% số điểm của cả test.
- Trong thời gian diễn ra vòng thi, chương trình của bạn sẽ được chấm trên test ví dụ của đề bài. Nếu bạn sai ở test này thì kết quả sẽ là "Wrong Answer". Ngược lại, dù bạn được trọn điểm hay một phần điểm, kết quả vẫn được hiển thị là "Accepted".