**Bài tập queue và stack**

Do Stack và queue khá giống nhau nên lần này bài tập sẽ ít thôi. Các bạn luyện tập để dùng thành thạo các phương thức của 2 lớp này nhé.

Bài 1: Xuất nội dung của queue sau mỗi thao tác trong dãy.

Một chữ cái tượng trưng cho thao tác thêm chữ cái đó vào trong queue, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong queue in lên màn hình.

Cho biết kết quả xuất ra màn hình sau khi hoàn tất chuỗi input.

Ví dụ:

input:

AB\*

ouput:

A

Bài 2: Tèo và các bạn đang học về cấu trúc dữ liệu Stack. Ban đầu chúng ta có một stack rỗng. Với Q truy vấn, mỗi truy vấn sẽ là một trong 3 dạng sau:

* 1 x   - thêm phần tử x vào đỉnh stack
* 2     - xóa phần tử ở đỉnh stack ra khỏi stack
* 3     - xuất ra màn hình phần tử lớn nhất đang có trong stack

Hãy viết chương trình thực hiện các truy vấn trên.

**Input:**

* Dòng 1: Số nguyên dương Q - số lượng truy vấn (1 <= Q <= 105)
* Q dòng tiếp theo, mỗi dòng là một truy vấn (theo mẫu một trong 3 dạng ở trên, x <= 10^9).

Dữ liệu đảm bảo tính hợp lý ở mỗi truy vấn.   
Ví dụ: không tồn tại truy vấn 2 hoặc 3 khi stack đang rỗng, v...v

**Output:**

Với mỗi truy vấn loại 3, in trên một dòng phần tử có giá trị lớn nhất đang có trong stack.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **10 1 7 2 1 2 2 1 6 1 2 2 3 1 9 3** | **6**  **9** |

Bài 3: (Khó hơn 1 tý tẹo) Tèo và các bạn đang học về cấu trúc dữ liệu Stack. Ban đầu chúng ta có một stack rỗng. Với Q truy vấn, mỗi truy vấn sẽ là một trong 3 dạng sau:

* 1 x   - thêm phần tử x vào đỉnh stack
* 2     - xóa phần tử ở đỉnh stack ra khỏi stack
* 3     - xuất ra màn hình phần tử thứ 2 tính từ đỉnh của stack

Hãy viết chương trình thực hiện các truy vấn trên.

**Input:**

* Dòng 1: Số nguyên dương Q - số lượng truy vấn (1 <= Q <= 105)
* Q dòng tiếp theo, mỗi dòng là một truy vấn (theo mẫu một trong 3 dạng ở trên, x <= 10^9).

Dữ liệu đảm bảo tính hợp lý ở mỗi truy vấn.   
Ví dụ: không tồn tại truy vấn 2 hoặc 3 khi stack đang rỗng, v...v

**Output:**

Với mỗi truy vấn loại 3, in trên một dòng phần tử thứ 2 tính tử đỉnh stack.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **9 1 7 2 1 2 2 1 6 1 2 3 1 9 3** | **6**  **2** |

Bài 4: **Xâu ngoặc đúng: (Bài này khó, Bài này tương tự với bài 1(Tức là bài dễ nhất) đề thi loại Samsung vòng online năm ngoái. Người ta đã gợi ý la dùng stack nhé.)**

// các bạn có thể làm bài này hoặc không làm. Ai làm được tất nhiên sẽ rất được bọn mình để ý.

Gọi xâu chỉ chứa các kí tự ngoặc tròn (, ), ngoặc vuông [, ] và ngoặc nhọn {, } là xâu ngoặc. Xâu ngoặc đúng được định nghĩa như sau:  
- Xâu rỗng được coi là xâu ngoặc đúng  
- Nếu a là xâu ngoặc đúng thì (a), {a}, [a] cũng là xâu ngoặc đúng  
- Nếu a và b là các xâu ngoặc đúng thì ab cũng là xâu ngoặc đúng

Viết chương trình xác định xâu đầu vào có phải xâu ngoặc đúng hay không?

**input:** nằm trong file ngoacdung.inp

Chứa một xâu cần xác định

**output:** xuất ra file ngoacdung.out

Nếu xâu ngoặc đúng thì xuất  “yes” ngược lại xuất ra “no”

**ví dụ**:

**input**:

[()]

**output:**

yes