Problem 263: Workflow Tracking

Difficulty: Medium

Author: Michael Wells, Grand Prairie, Texas, United States

Originally Published: Code Quest 2025

Problem Background

Managing software is a difficult process; in large projects, it can be difficult to add new features or update existing ones without introducing new bugs or otherwise breaking other features. Simply adding new features without considering the impacts can lead to significantly higher costs and delivery times. Lockheed Martin often uses a process managed by a "Change Review Board" to ensure requests for new features are fully understood and evaluated.

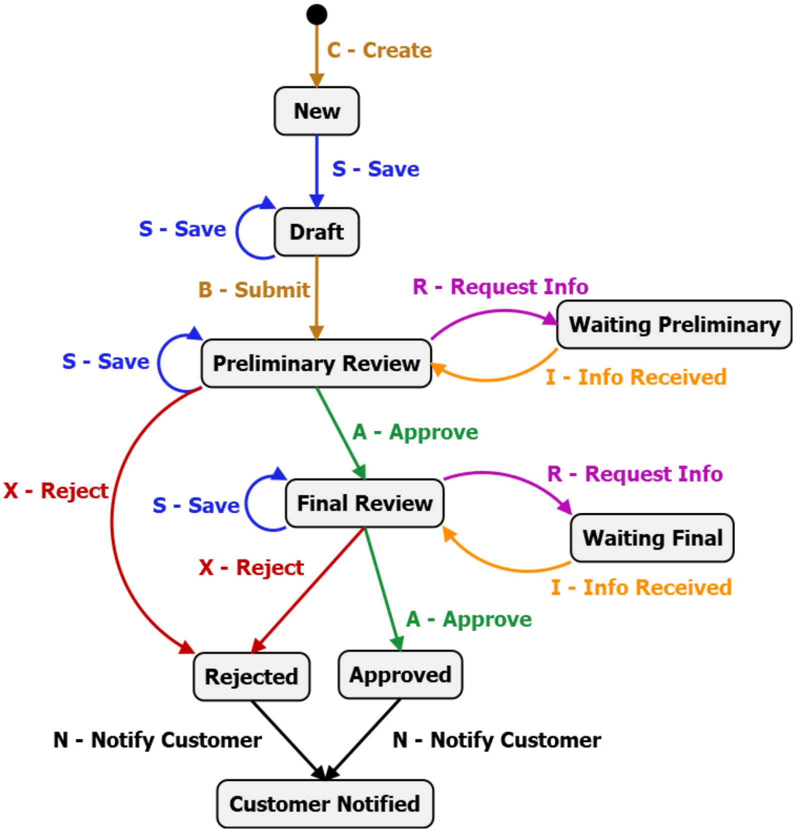
You've been appointed as the lead for your project's Change Review Board, and would like to streamline things by creating an automated process to handle requests from the customer.

Problem Description

Your project's Change Review Board generally follows the process shown in the workflow diagram at right.

A change request goes through a series of states (from top to bottom) as it's reviewed by the Change Review Board. Each of the boxes in the diagram shows a state that a request could be in at any given time; each arrow shows how the request can move from one state to another, and why. You'll need to create a program that implements this workflow and can validate if a request is being processed correctly.

For example, if a request is in the "Final Review" state, a reviewer could "Approve" or "Reject" it; they cannot "Submit" the request, as there's no arrow coming from "Final Review" with that label.



The rest of your team is working on creating an interface for this tracking tool; they'll develop a menu that provides controls for each of the actions (arrows) in the diagram above. These will be identified in the software using single-character codes, as follows:

- C - Create
- S - Save
- B - Submit
- R - Request Info
- I - Info Received
- X - Reject
- A - Approve
- N - Notify Customer

Your task is to work on creating a program that can validate that these actions are being executed correctly. You'll be given a string showing a list of command codes. You must output what state the request is in following each command. If a given command is invalid for the request's current state, don't change the request's state, but instead output "Invalid Command".

Sample Input

The first line of your program's input, **received from the standard input channel**, will contain a positive integer representing the number of test cases. Each test case will include the following information, separated by spaces:

- A positive integer, representing the request's tracking ID
- Two or more uppercase letters, from the list provided above, representing commands that were attempted in relation to this request, in the order they were attempted.

```
4
1 C S B R I A X N
2 C B S B A A N
3 S C S B X N A
4 C S B A R I A N
```

Sample Output

For each test case, your program must print a single line containing:

- The tracking ID for the request
- A space
- For each command issued for the request:
  - If the command was not the first issued for the request, a 'greater-than' sign (>)
  - The name of the state resulting from execution of the command (as provided in the diagram above), or "Invalid Command" if the given command is not valid for the current state of the request

Note that due to the length of each line of output, the lines in the sample provided below are breaking across multiple lines within this document. Please download the sample output file from the contest website for a more accurate depiction of the output. Each test case's output consists of a single line, starting with an integer.

```
1 New>Draft>Preliminary Review>Waiting Preliminary>Preliminary Review>Final
Review>Rejected>Customer Notified
2 New>Invalid Command>Draft>Preliminary Review>Final Review>Approved>Customer
Notified
3 Invalid Command>New>Draft>Preliminary Review>Rejected>Customer Notified>Invalid
Command
4 New>Draft>Preliminary Review>Final Review>Waiting Final>Final
Review>Approved>Customer Notified
```