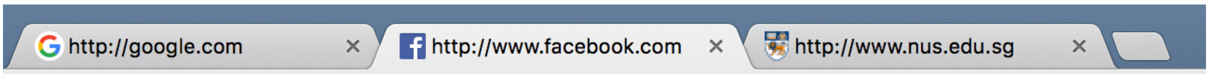
Browser

Eugene is a university lecturer who is teaching a basic data structures and algorithms course. At the moment, Eugene is bored. Since he has nothing better to do now, he starts playing around with his internet browser. His browser, along with most browsers available today, has a support for multiple tabs. This means that we can open multiple websites in one single window, in a mini-window that we call 'tabs'. At any given time, there is at least one tab open. The tabs are arranged in a list. One of them is a 'current tab' that the user is currently viewing. Here is an example of tabs:



There are three tabs open in the picture above, and the active tab is "http://www.facebook.com". We say that the tab "http://www.facebook.com" is located next to the tab "http://google.com" and the tab "http://www.nus.edu.sg" is located next to the tab "http://www.facebook.com".

Suddenly, when Eugene is browsing through the many tabs in his web browser, he realizes that he can make this into a simple exercise for his students. Hence, he began formulating this simple exercise for his students to do:

1. When a user opens the browser, the page "**http://www.comp.nus.edu.sg**" is opened by default.
2. There are multiple operations that users can do in the browser:
   - Open a new tab that opens the page "**http://www.comp.nus.edu.sg**".
   - Close the current tab.
   - Switch to the next tab in the list.
   - Switch to the previous tab in the list.
   - Open a particular page in the current tab.
   - Open a new tab that opens a specific page.

Before releasing this assignment to his students, Eugene asked you, his colleague, to test whether all the sample input and outputs are correct. Help Eugene prepare this assignment for his students.

Input

The first line of input contains an integer **N** (1 <= N <= 100), the number of user operations. **N** rows follow, each containing a single query for you to answer. The queries follow the following specification:

Query Type Input Format: `<QUERY_TYPE> <APPROPRIATE_PARAMETERS>`

1. `NEWTAB`
   Create a new tab that opens the default page as stated in the problem description above next to the current tab.

2. `CLOSETAB`
   Close the current tab. The current tab moves to the next tab. If the closed tab is the last tab in the list, move to the previous tab instead. It is guaranteed that there are at least two open tabs when this query comes.

3. `NEXTTAB`
   Switch the current tab to be the next tab. If the current tab is the last tab, do nothing.

4. `PREVTAB`
   Switch the current tab to be the previous tab. If the current tab is the first tab, do nothing.

5. `OPENHERE URL`
   Open the page URL at the current tab.

6. `OPENNEW URL`
   Open the page URL in a new tab located next to the current tab.

Output

For each query, print the URL of the current tab. The last line of output contains a newline character.

Sample Input

```
10
OPENNEW http://www.google.com
OPENHERE http://www.facebook.com
NEXTTAB
NEWTAB
NEXTTAB
NEXTTAB
CLOSETAB
PREVTAB
CLOSETAB
OPENNEW http://ivle.nus.edu.sg
```

Sample Output

```
http://www.comp.nus.edu.sg
http://www.facebook.com
http://www.google.com
http://www.google.com
http://www.comp.nus.edu.sg
http://www.comp.nus.edu.sg
http://www.google.com
http://www.facebook.com
http://www.google.com
http://www.google.com
```

Explanation

These are the tabs open and the current tab (bolded and underlined) after each query. The first line is the initial state when a user opens the browser:

http://www.comp.nus.edu.sg
http://www.comp.nus.edu.sg | http://www.google.com
http://www.facebook.com | http://www.google.com
http://www.facebook.com | **http://www.google.com**
http://www.facebook.com | **http://www.google.com** | http://www.comp.nus.edu.sg
http://www.facebook.com | http://www.google.com | **http://www.comp.nus.edu.sg**
http://www.facebook.com | http://www.google.com | **http://www.comp.nus.edu.sg**
http://www.facebook.com | **http://www.google.com**
http://www.facebook.com | http://www.google.com
http://www.google.com
http://www.google.com | http://ivle.nus.edu.sg

Skeleton

You are given the skeleton file Browser.java (see contents on file).

Notes

1. You **must use linked list** to solve this problem.
2. You are free to define your own linked list class (encouraged for practice and skeleton file given), but you are allowed to use Java's built-in linked list implementation if it is suitable for this problem.
3. You are free to (and should) modify the skeleton file and add more attributes or methods when necessary.