

Sam is developing a new web browser and he feels that back, forward and refreshes buttons are too boring, he feels showing the URL of the website which they point to would be a better approach, he is facing trouble with this task and might be fired from his job all because he didn't study an important course well in college, you being a student of data structure and algorithms decide to help your friend.

Following operations are possible

V <URL> : Visit the input URL

B <N> : Back N steps, in case not possible print "m back not possible", then print as stated

F <N> : Forwards N steps print "m forward not possible", then print as stated

Q <N> : Shows URL N step from the present URL (N is positive for forward and negative for backward, in case N is out of bound print None)

D <N> : Delete Nth step URL from the present URL (N is positive for forward and negative for backward, in case N is out of bound print None)

E : Exit

With each operation print (except Q and D) the URL of one step back, present (refresh) and one step forward.

Assumptions and Clarifications:

1 - Browser has only 1 tab

3 - Back and Forward take place one step at a time, for example, if you want to go back 3 steps, it would mean going back 2 steps then taking one more step backwards and so on.

4 - If no URL is present while front or back, you need to display "None"

Important Note: It is compulsory to use a doubly-linked list for this assignment as your main data structure.

Sample Input:

V google.com

V twitter.com

V google.com

B 1

F 2

Q -2

D -1

B 1

V linkedin.com

B 1

E

Output

None google.com None
google.com twitter.com None
twitter.com google.com None
google.com twitter.com google.com
1 forward Not possible
twitter.com google.com None
google.com
None google.com google.com
google.com linkedin.com None
None google.com linkedin.com

Explanation:

V google.com -

Doubly linked list:
[google.com]*

Print:
None google.com None

V twitter.com -

Doubly linked list:
[google.com] <-> [twitter.com]*

Print:
google.com twitter.com None

V google.com -

Doubly linked list:
[google.com] <-> [twitter.com] <-> [google.com]*

Print:
twitter.com google.com None

B 1

Doubly linked list:
[google.com] <-> [twitter.com]* <-> [google.com]

Print:
google.com twitter.com google.com

F 2

Doubly linked list:
After 1 forward
[google.com] <-> [twitter.com] <-> [google.com]*
Remaining 1 forward not possible

Print:
1 Forward Not possible
twitter.com google.com None

Q -2

Doubly linked list:
[google.com] <-> [twitter.com] <-> [google.com]*

Print:
google.com

D -1

Doubly linked list:
[google.com] <-> [google.com]*

B 1

Doubly linked list:
[google.com]* <-> [google.com]

Print:
None google.com google.com

V linkedin.com

Doubly linked list:
[google.com] <-> [linkedin.com]*

Print:
google.com linkedin.com None

B 1

Doubly linked list:
[google.com]* <-> [linkedin.com]

Print:
None google.com linkedin.com

E

Code terminates