Intermediate Division Programming Problem: Duplicates

PROBLEM: Consider a sorted array containing the letters ABRACADABRACABOB:

A	A	A	A	A	A	B	B	B	B	C	C	D	O	R	R	

The array at this point is 16 elements long, but there are many duplicate elements. We can save space if we keep a single copy of each letter (the array "LETTERS"), and also an auxiliary array with the count of the number of times that each letter appears (the array "COUNTS"):

LETTERS	A	B	C	D	O	R	
COUNTS	6	4	2	1	1	2	

There is some extra overhead in maintaining the auxiliary array. When a letter that you haven't seen before is added, a new slot must be made in both arrays; when you add a letter that is already in the array of letters, the counter in the auxiliary array is incremented; and finally, when you delete a letter, you decrement the counter in the auxiliary array. However, when the counter hits zero, you remove the item from the LETTERS array and you remove the corresponding entry in the COUNTS array.

Continuing with the example above, let's add the letters BATH:

Add B:

LETTERS	A	B	C	D	O	R	
COUNTS	6	5	2	1	1	2	

Add A:

LETTERS	A	B	C	D	O	R	
COUNTS	7	5	2	1	1	2	

Add T:

LETTERS	A	B	C	D	O	R	T	
COUNTS	7	5	2	1	1	2	1	

Add H:

LETTERS	A	B	C	D	H	O	R	T	
COUNTS	7	5	2	1	1	1	2	1	

Now, if we delete the letters BOA, the arrays change as follows:

Delete B:

LETTERS	A	B	C	D	H	O	R	T	
COUNTS	7	4	2	1	1	1	2	1	

Delete O:

LETTERS	A	B	C	D	H	R	T	
COUNTS	7	4	2	1	1	2	1	

Delete A:

LETTERS	A	B	C	D	H	R	T	
COUNTS	6	4	2	1	1	2	1	

INPUT: You'll process 4 commands: RESET, ADD, DELETE, and REPORT.

- The "RESET *xxx*" command clears the arrays and also erases the history of what happened in each position of the arrays. Then, adds the letters in *xxx*, one at a time.

- The "ADD *xxx*" command adds the letters in *xxx*, one at a time.
- The "DELETE *xxx*" command removes the letters in *xxx*, one at a time. If a letter is not in the array of letters, just ignore it.
- The "REPORT *N*" command requires you to print the historical contents of the *N*th element of the LETTERS array each time it changes, since the last time a RESET command was processed.

For example, the commands "RESET abracadabracabob", "ADD bath", "DELETE boa" results in the arrays show above. "REPORT 3" has a value of "RC" because element 3 of the LETTERS array had an R, and later contained a C.

In the RESET, ADD, DELETE commands, ignore all non-letters in the string and convert all lower-case letters to uppercase. A RESET command will be the first input line.

OUTPUT: After each REPORT command, print a string that is the elements of the Nth position of the array over time. There will be 5 REPORT commands

SAMPLE INPUT

```
RESET abracadabracabob
REPORT 3
REPORT 5
ADD BATH
DELETE boa
REPORT 5
DELETE drr
REPORT 5
RESET American Computer Science League
ADD Computer
DELETE Computer
DELETE COMPUTER
REPORT 10
```

SAMPLE OUTPUT

```
1. RC
2. RO
3. ROH
4. ROHRT
5. UTSRPRS
```

TEST DATA

TEST INPUT

```
RESET simple simon
REPORT 4
ADD simply said something slowly
REPORT 4
DELETE so say something
REPORT 4
RESET peter piper picked a peck of pickled
REPORT 7
DELETE pickled
DELETE sunflowers
ADD pickled
ADD sunflowers
REPORT 5
```

TEST OUTPUT

1. SPM
2. SPMLIHG
3. SPMLIHGHIL
4. TRPOK
5. TRPKIFIF