

Program Name: `scoring.cpp`Input File: `scoring.dat`

In a hypothetical programming contest all judging activity is logged, and those logs are then used to verify the outcome of the contest. Teams are ranked by a final score determined by the number of problems solved and the number of rejections. You are given the following assumptions for this problem:

1. There are 10 teams in the contest. The judges and scorers use only the team numbers (1-10) to for identification to preserve anonymity and guarantee the most accurate and fair outcome.
2. There are 6 problems in the problem set.
3. Each problem in the problem set has its own point value.
4. A rejection by a team on a problem costs a point against the problem's point value **only** if the problem is solved.
5. The minimum score awarded for a problem that is solved is 1 point even if the rejection penalties for the problem exceed the point total for the problem.
6. Teams are ranked by the final score (highest to lowest).
7. If two or more teams have the same final score, then the following tiebreakers are applied to determine rank:
 - The team with the fewest rejections (regardless of whether the problem was accepted) wins.
 - If the number of rejections is the same for two teams who have the same final score, then the team who had a problem accepted first wins.

Your program will read a contest definition along with the contest logs and determine the final rankings for all 10 teams in the contest.

Input

Input to your program consists of a contest definition followed by the contest logs. The first line of the input file contains the contest definition and consists of a series of integer scores ($1 \leq S_p \leq 6$) where the integers represent the point value of the problems in order (1-6). Each of the remaining input lines represent a judging action. Each judging action has the following format:

1. the result of the judging action ('A' for accept or 'R' for reject) and appears in column 1
2. a single space
3. the team number ($1 \leq T \leq 10$) for the judging action
4. a single space
5. the problem number being judged ($1 \leq P \leq 6$)

You can be assured that there will be no extraneous input including blank lines or trailing characters at the end of a line. You can also be assured that there will be a maximum of 1 acceptance judging line for a given team number/problem number combination. Your program should read to the end of file processing all judging actions before determining the final team rankings.

Output

The output from your program consists of a single line in which you will print the team numbers in rank order for the teams who solved at least one problem. Do not print anything for the teams that solved zero problems. The integer team numbers should be listed on a single line in order of rank starting with the winning team with exactly one blank space between each team number.

Example: Input File

4 8 2 4 10 1

A 1 3

A 4 1

R 10 5

A 2 3

A 8 1

A 7 4

R 5 6

R 7 5

R 2 6

R 7 5

A 10 4

R 4 2

R 1 4

A 5 3

R 7 2

R 5 6

A 2 4

R 10 2

A 1 4

R 4 4

A 7 5

R 5 6

A 4 3

R 4 2

R 8 3

A 4 4

R 2 6

R 10 2

A 7 1

A 5 6

R 10 2

R 4 2

A 7 2

R 4 2

A 8 3

A 10 2

Output to screen

7 10 4 2 1 8 5