

11537 Secret Problemsetters Union

Solving problem is tough? Solving problem is easy? Rather confusing. But no doubt, problem solving is fun. Again, there's no doubt that problem setting is hard. Let alone the art of finding new ideas & all those statement, solution, data stuffs. The toughest part is to live with the fact that, if ever there's the slightest of errors, you are doomed. Hence, setting quality problems requires time. But once upon a time in a far far away country, contest organizers forgot this fact completely. So they went on organizing one after another prestigious contests with very short notice to the problem setters. However, the contests had grand success & everyone forgot about what happened to the poor problem setters thereafter. Hence, the problem setters united & decided to have a useful plan for the future.



The problem setters decided that everyone of them will create problems throughout the year & maintain his own problem bank. Whenever someone creates a new problem, he adds the problem to his problem bank & the problem setting director gives it a rating. Rating is an integer between 1 and 100007. Whenever a problem setter is requested to arrange a contest, best K problems (K problems with maximum rating) are picked from his bank. The problems are used & removed from the bank. There can be at most 100 problems in a contest.

Once a year, before the tide of contest season begins, the problem setting director collects all the problems from the available banks & merges into a central problem bank. Any subsequent request for problem is served from the central problem bank.

You are to write a program that manages the problem bank.

Input

Input file start with an integer C ($1 \leq C \leq 15$), The number of test cases. Every test case starts with an integer P ($1 \leq P \leq 50$) indicating the number of problems setters for that test case. Every problem setter has his own problem bank. The bank of problem setter X is denoted by the integer X ($1 \leq X \leq P$). At the beginning of a test case, all problem banks are empty. Another important thing to notice is that, at a time there will be at most 5000000 problems in the merged problem bank & at most 100000 problems in any other bank. Rest of the lines describe the operations on a problem bank using some instructions. There can be 4 kinds of instructions as described below:

- An 'I' instruction stands for an insert operation.

The format of a single character 'I' followed by 4 integers B ($1 \leq B \leq P$), N ($0 \leq N \leq 100000$), A_0 ($0 \leq A_0 \leq 1000000$) & X ($0 \leq X \leq 10000000$), means insert N problems in the problem bank with id B . The ratings of these N problems can be calculated using A_0 & X . The rating of a particular problem $A_n = (A_{n-1} + n * X) \% 100007$ where $n = 1$ to N .

- An 'U' instruction stands for a use & remove operation.

The ‘U’ instruction consists of a single ‘U’ character followed by two integers B ($1 \leq B \leq P$) & K ($0 \leq K \leq \text{number_of_problems_in_bank}B$) meaning that pick & remove the best K problems from problem bank B .

- A ‘M’ instruction means a merge operation.

There’ll be at most one merge operation in a test case. A ‘M’ instruction merges all the existing problem banks in a single problem bank. This merged bank is denoted as bank #1 thereafter.

- A ‘Q’ instruction indicates the end of a test case.

Output

For every ‘U’ instruction you receive, print 2 integers in a line showing the highest & lowest rating of the problems selected with that particular instruction. If you don’t pick any problem print a couple of ‘0’. Print a blank line between test cases.

Explanation of Sample Input:

As we see, here are 3 problem setters in this case. We’ll call their problem banks 1, 2 & 3. The first I instruction adds 3 problems to bank 1. The ratings of these 3 problems are, 1 ($0 + 1 * 1$), 3 ($1 + 2 * 1$) & 6 ($3 + 3 * 1$). The second I instruction adds 5 problems to bank 2 with ratings 3 ($0 + 1 * 3$), 9 ($3 + 2 * 3$), 18 ($9 + 3 * 3$), 30 ($18 + 4 * 3$) & 45 ($30 + 5 * 3$). The third I instruction adds 4 problems to bank 3 with ratings 5 ($2 + 1 * 3$), 11 ($5 + 2 * 3$), 20 ($11 + 3 * 3$) & 32 ($20 + 4 * 3$). The next I instruction adds 2 more problems to bank 1. The ratings are 8 ($5 + 1 * 3$) & 14 ($8 + 2 * 3$). That means, bank 1 now has 5 problems. Then comes an U instruction. The first U instruction picks the 4 most rated problems from bank 1 i.e. the problems with rating 14, 8, 6 & 3. You should print 14 & 3. The next U instruction selects 2 best problems from bank 2. They are the problems with rating 45 & 30. The merge operation merges the remaining problems (1, 3 & 4 problems from bank 1, 2 & 3 respectively) into a single bank & names it bank 1. The following U instruction is served from this merged bank & the problems with rating 32, 20 & 18 are selected. The final I instruction asks to add 0 new problems to the merged problem bank.

Sample Input

```
1
3
I 1 3 0 1
I 2 5 0 3
I 3 4 2 3
I 1 2 5 3
U 1 4
U 2 2
M
U 1 3
I 1 0 1 1
Q
```

Sample Output

```
14 3
45 30
32 18
```