Ranking Prediction

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

You and your teammates have just finished an ICPC competition. Five hours of competition have drained your energy, and your teammate has eaten your lunch. Now you can only lean on the desk and look at the ranking board. The award ceremony has not yet taken place, so the ranking is still frozen, meaning that you know the submission times and whether your team passed during the entire competition, but for other teams, you know the time of each submission they made, and you know whether each submission was accepted before the ranking was frozen, but you do not know whether the submissions made after the ranking was frozen were accepted.

When you check the ranking, you notice a team you are very concerned about. You know the time and status of each submission they made before the ranking was frozen, as well as the times of each submission made after the ranking was frozen. You want to know whether their team's ranking will be **strictly higher** than your team's. To determine the possibility of their ranking being strictly higher than yours, you also want to know the minimum number of problems they need to solve after the ranking is frozen.

In ICPC competitions, the penalty time is calculated according to the following rules. Suppose a team **solved** m problems, numbered from 1 to m. For each solved problem i, let the time of the first accepted submission be t_i, and the number of submissions before solving this problem be c_i. The penalty time p is calculated as follows:

$$p = \sum_{i=1}^{m} t_i + 20 \cdot c_i$$

In this problem, special factors such as compile errors that do not count towards penalty time are not considered.

For two teams A and B, team A is said to have a strictly higher ranking than team B if and only if the number of problems solved by A is greater than that solved by B, or if the number of problems solved by A and B is equal, and the penalty time of A is less than that of B.

Input

The first line contains an integer T ($1 \le T \le 100$), indicating the number of test cases.

For each test case, the first line contains three integers n, a, b ($10 \le n \le 15, 1 \le a \le n, 0 \le b \le 10^5$), representing that there are n problems in the competition, your team solved a problems by the end of the competition, and the penalty time is b.

The second line contains an integer s ($0 \le s \le 10^3$), indicating the number of submissions made by the team you are concerned about during the normal competition.

The next s lines each contain an integer followed by two strings t, p, v ($0 \le t < 300$). This indicates that at minute t, a submission was made for problem p, and the result was v. It is guaranteed that submissions are given in chronological order, p is one of the first n uppercase letters, and $v \in \{\texttt{ac}, \texttt{rj}, \texttt{pd}\}$. `ac` means this submission was accepted, `rj` means the submission was rejected, and `pd` means this submission is in a frozen ranking state, and it is unknown whether this submission was accepted. It is guaranteed that when $t < 240$, v is not `pd`, and when $t \ge 240$, v is guaranteed to be `pd`.

Output

For each test case, output a single integer on a new line. If it is impossible for the team you are concerned about to have a final ranking strictly higher than your team, output -1; otherwise, output the minimum number of problems they need to solve after the ranking is frozen for their final ranking to be strictly

higher than yours.

Example

standard input	standard output
1 11 6 900 13 11 C ac 34 J ac 52 D rj 61 D ac 193 A rj 207 A rj 220 G ac 245 A pd 247 A pd 262 H pd 299 A pd 299 C pd 299 K pd	2