

Killing The Time

Input file: **stdin**
Output file: **stdout**
Time limit: 10 seconds
Memory limit: 512 megabytes

Roma has downloaded a modern strategic game with RPG elements on his smartphone. In the game, the player is using a turn-based mechanism to command three characters and fight the hordes of enemies. Since the game is a shareware and Roma is a serious guy that does not spend money on games, this game is available to him in a very limited mode.

The number of moves is severely limited, enemies do not move, enemies and characters do not die, but the score at the end of the game is still displayed. As Roma has figured out from his attempts to play this game, the score consists of the sum of damage dealt by characters to the enemies and life healed to the characters minus the sum of damage dealt by the enemies to the characters and life healed by the characters to the enemies. Roma is about to set the absolute record for each of the levels. And you are about to help him!

The game field is a rectangular tiled grid of size $N \times M$. Some tiles are occupied by enemies, some other by impassable obstacles, yet some other by three Roma's characters, and all other tiles are free. The game process is working in the following way. Roma moves first, then moves the almighty artificial intelligence. In his turn, Roma chooses the order in which the characters will move, and then makes those moves sequentially, character by character. During his turn, Roma can move the character and then apply one of its abilities. The characters move by steps (one tile by each step) either horizontally or vertically so that the total number of moves does not exceed that character's stamina. Characters can not pass through the impassable obstacles, enemies or other characters. Roma can also decide not to move the character at all, or not to apply any of its abilities, or not to do anything by that character at all. After Roma's turn, the enemies take action, and artificial intelligence tries to minimize the player's score. Then a new turn begins or the game ends.

Each of the characters' abilities is one of the following:

- “TELEPORT L ”: the character can teleport to any free tile of the game field with Manhattan distance not more than L from the current tile.
- “SWAP L ”: the character can exchange places with any of the other characters with Manhattan distance not more than L from the current tile.
- “SINGLE ATTACK $r R d$ ”: the character deals d base damage to any one selected tile with Manhattan distance not less than r and not greater than R from his current position.
- “SPLASH ATTACK $r R d$ ”: the character deals d base damage to all tiles with Manhattan distance not less than r and not greater than R from his current position.
- “SINGLE HEAL $r R d$ ”: the character heals d life to any one selected tile with Manhattan distance not less than r and not greater than R from his current position.
- “SPLASH HEAL $r R d$ ”: the character heals d life to all tiles with Manhattan distance not less than r and not greater than R from his current position.

Each type of enemy is one of the following:

- “SINGLE $r R d$ ”: the enemy deals d base damage to any one selected tile with Manhattan distance not less than r and not greater than R from its current position.
- “SPLASH $r R d$ ”: the enemy deals d base damage to all tiles with the Manhattan distance not less than r and not greater than R from its current position.

Each enemy has three multipliers for the dealt damage: it deals $d \cdot M_1$ damage to the first character, $d \cdot M_2$ to the second character and $d \cdot M_3$ to the third character. Additionally, each enemy has three multipliers for the damage received: damage from the first character is multiplied by S_1 , from the second by S_2 , and from the third by S_3 .

Since the game is played on hardcore level, the characters can deal damage to each other (base damage value) and also can heal enemies, but the enemies do not deal damage to each other.

The Manhattan distance between two positions is the sum of the absolute difference between row numbers of these two positions and of the absolute difference between column numbers of these two positions.

You are to calculate the maximum score that Roma can earn in the given game.

Input

The first line of input contains integers N , M , D and K : the dimensions of the game field, the length of the game and the number of types of enemies on the level K ($1 \leq N, M, D \leq 8$, $0 \leq K \leq 26$).

On the next N lines, the game field is given. Each of them contains exactly M characters: “.” denotes the field is empty; “#” denotes the field is not passable; “1”, “2” or “3” denotes the positions of the characters; “A”..“ α ” (where α is the K -th letter of English alphabet) denotes the type of the enemy on the field. It is guaranteed that there is exactly one of each character “1”, “2” and “3” on the field.

On the next K lines, the descriptions of the enemy types are given. First goes the type of damage, and then follow nine numbers r , R , d , M_1 , M_2 , M_3 , S_1 , S_2 , S_3 ($0 \leq r \leq R \leq 20$, $1 \leq d, M_1, M_2, M_3, S_1, S_2, S_3 \leq 1000$). The first description is for type “A”, the second one for type “B” and so on.

Then the descriptions of three characters are given. On the first line of each of these descriptions, two numbers are given: V , the stamina of the character and A , the number of his abilities ($0 \leq V \leq 5$, $0 \leq A \leq 100$). Then A lines describe the abilities in the format “name of ability L ” for movement abilities ($0 \leq L \leq 5$) and “name of ability $r R d$ ” for the other abilities ($0 \leq r \leq R \leq 20$, $1 \leq d \leq 1000$).

Output

Output one integer: the maximum score that Roma can earn in the given game.

Examples

stdin	stdout
4 4 1 112. AA3A SINGLE 0 1 10 1 1 100 10 10 1 1 2 SPLASH HEAL 0 3 4 TELEPORT 1 1 2 SINGLE ATTACK 0 10 10 SWAP 2 1 1 SPLASH ATTACK 1 1 1000	1984
1 6 8 2 123#BA SINGLE 5 5 10 1000 100 1 1 1 1 SINGLE 2 2 5 300 1 100 1 1 1 5 1 SINGLE ATTACK 5 5 1 5 1 SPLASH HEAL 1 1 1 5 2 SPLASH ATTACK 0 1 100 SWAP 2	-5574