Problem J. Jackpot

Input file:	`jackpot.in`
Output file:	`jackpot.out`
Time limit:	3.5 seconds
Memory limit:	512 mebibytes

Johnny is on vacation in Las Figas. He has decided to visit the new casino *Lohal* which has opened there recently. Lohal offers the new game to its visitors, called *Black or White*. It is a single player game that proceeds as follows.

There are m tokens arranged in a line. The player is offered tokens one after another. Each token is either white, black, or diamond. To get a token the player needs to buy a card. There are several types of cards available.

The card of the i-th type has two numbers written on it: w_i and b_i, there are c_i such cards. To buy the card of the i-th type the player must pay $w_i + b_i$ dollars, but he can get a discount if he already owns some tokens.

That is, if the player has a white tokens, he can decrease the number w_i on the card by at most a (but he cannot make it negative). Similarly, if the player has b black tokens, he can decrease the the number b_i by at most b (but still cannot make it negative). Finally, for each diamond token the player can decrease by 1 any of w_i or b_i of his choice.

For example, if the player has 2 white, 1 black and 1 diamond token, he can buy the card that has $w_i = 4$ and $b_i = 0$ for 1 dollar: 2 white tokens decrease w_i to 2, using diamond token allows to decrease w_i to 1. Black token is useless for this card, so its cost now is $1 + 0 = 1$. The same set of tokens allows to get the card that has $w_i = 2$ and $b_i = 2$ for free (now we use diamond token to decrease b_i).

Note that buying a card doesn't require the player to discard the tokens, they stay at him and can later be used again to buy other cards.

The player can choose which tokens to get, but to win a jackpot he needs to get all tokens. Johnny wants to win a jackpot so he decided to get all tokens. Help him to choose which cards to buy to get tokens, so that he wins jackpot spending minimal possible sum.

Input

The input file contains several test cases.

Each test case starts with an integer m — number of tokens ($1 \le m \le 300$). The following line contains m characters, each of them is either '`W`' for white token, '`B`' for black token, or '`*`' for diamond token. The line lists tokens in order they are offered to the player.

The following line contains n — the number of card types ($1 \le n \le 300$). The following n lines contain three integers each: for each card type w_i, b_i and c_i are specified ($0 \le w_i, b_i \le 1000$, $1 \le c_i \le 300$). The sum of all c_i is at least m.

The last test case is followed by a line that contains 0. It must not be processed.

The sum of m for all test cases in one input file doesn't exceed 300. The sum of n for all test cases in one input file doesn't exceed 300.

Output

For each test case output one integer: the minimal sum Johnny must spend to win a jackpot.

Examples

jackpot.in	jackpot.out
5 WWB*W 3 0 1 1 1 0 1 1 2 3 0	4

The optimal strategy in the example is, for example, the following.

Token	**Card type**	**Player has**	**Cost**
W	1		$0 + 1 = 1$
W	2	W	$(1 - 1) + 0 = 0$
B	3	WW	$(1 - 1) + 2 = 2$
*	3	WWB	$(1 - 1) + (2 - 1) = 1$
W	3	WWB*	$(1 - 1) + (2 - 2) = 0$ (use diamond to decrease b_3)