GALAWAGA

Description

Let S be a string consisting of uppercase letters, and let T be one of its substrings. Let S be a string that results from removing an instance of T from S. For instance, suppose that S and T are equal to *GALAWAGAWA* and *WA*, respectively. Then, S could be either *GALAWAGA* or *GALAGAWA*.

By setting S equal to S, this process can be repeated a finite number of times. Moreover, the final string S will be unique and will not contain T. Determine the final string S.

Input

The first line will contain the number of test cases, n, that must be processed ($1 \leq n \leq 100$). The next n lines will each contain two strings, S and T ($1 \leq |S|, |T| \leq 100$). All strings will consist of uppercase letters.

Output

Print the result of each test case on a separate line.

Sample Input

```
3
GALAWAGA WA
GALAWWAAGWAA WA
NOTGALAGA WA
```

Sample Output

```
GALAGA
GALAGA
NOTGALAGA
```

Pancakes

Description

Timmy is currently planning to diet. However, he does not want to stop eating his favorite food: pancakes! Timmy is designing a schedule for the n meals that he will eat during his diet. In order to ensure that his pancake-cravings are satisfied, every continuous subsequence of L meals will contain at least one pancake meal.

For example, if $n = 6$ and $L = 3$, the following diets would be valid and invalid, respectively:

sandwich pancakes sandwich salad pancakes pancakes

sandwich pancakes sandwich salad potatoes pancakes

Timmy has almost finished planning his meal schedule, but is now requiring a bit of help. Taking his incomplete schedule into account, Timmy would like to minimize the total amount of pancake meals during his diet. If this number is too high, Timmy will need to modify his current plan. Timmy has requested your assistance to determine the minimum number of unplanned meals that must be converted into pancake meals, such that he will eat pancakes at least once during every L consecutive meals.

Input

The first line will contain the number of test cases, T, that must be processed ($1 \leq T \leq 500$). Then, T test cases will follow in the following format:

- The first line of each test case will contain two integers, n and L. The number n will denote the number of meals that Timmy will eat during his diet ($1 \leq n \leq 100000$). Additionally, Timmy would like to eat pancakes at least once during each L consecutive meals ($1 \leq L \leq n$).
- The second line will contain n space-separated strings; these strings represent Timmys current diet plan. The i-th string, M_i, will consist of either only lowercase letters ($1 \leq |M_i| \leq 10$), or it will be a single question mark. A pancake meal will be denoted by the string, "pancakes" (without quotes). A single question mark will denote an unplanned meal.

Output

Print the minimum amount of unplanned meals that must be converted into pancake meals. If Timmys incomplete diet plan will not allow him to eat pancakes at least once during each span of L consecutive meals, print "-1" (without quotes). The result of each test case must be printed on its own line.

Sample Input

3
6 3
oatmeal pancakes veggies ? ? pancakes
6 3
pancakes oatmeal ? pancakes tofu veggies

6 3
pancakes tofu ? veggies oatmeal smoothie

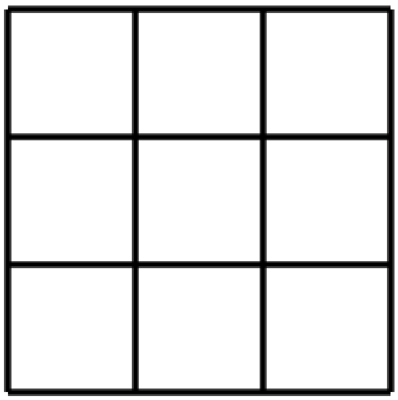
Sample Output

1
0
-1

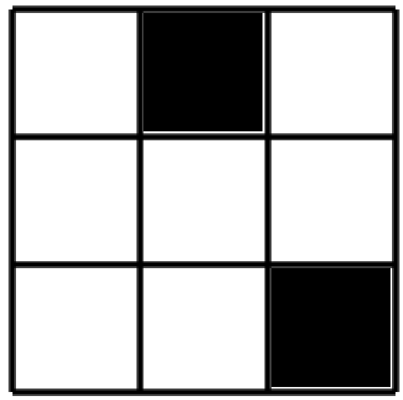
Perfect Boards

Description

An n by m board is called perfect if all of its cells are of the same color. Each cell of the board is a 1×1 square, and is either black or white.

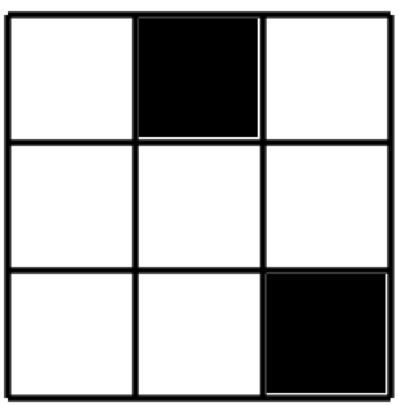


A perfect board

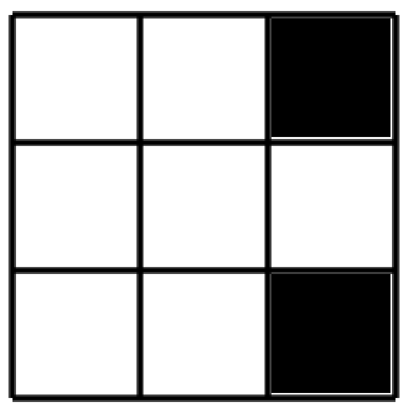


An imperfect board

Two cells are adjacent if they can be covered completely by a 1×2 or 2×1 rectangle. At any moment, exactly two adjacent cells can have their colors inverted. This process will cause a black cell to become white, and a white cell to become black. For example, two cells in the first row were inverted below.



Original board



Board after inverting a 1×2 rectangle

The cost for inverting a 1×2 rectangle is C_1 units while the cost for inverting a 2×1 rectangle is C_2 units. Determine the minimum cost for converting a given board into a perfect board. Note that a given board cannot be rotated.

Input

The first line of the input will contain the number of test cases, T, that must be processed ($1 \leq T \leq 10000$). Then, T test cases will be listed in the following format:

- The first line of each test case will contain two integers, n and m ($1 \leq n, m \leq 4$).
- The second line will contain the integer-valued costs C_1 and C_2 ($1 \leq C_1, C_2 \leq 10^7$).
- Lines 3 through $n+2$ will each contain m characters. The j-th character of the i-th row, $b_{i,j}$, will represent the cell located at the i-th row and j-th column of the board. The value of each $b_{i,j}$ will be either **B** or **W** to denote either a black or white cell, respectively.

Output

Print the minimum cost for converting the given board into a perfect board. If it is impossible to convert the given board into a perfect board, print -1 (without quotes). Each result must be printed on its own line.

Sample Input

```
2
3 3
2 3
BBW
BBB
BBB
4 4
2 3
BBBW
BBWB
BBBB
WBWB
```

Sample Output

```
9
9
```