USACO 2026 FIRST CONTEST, BRONZE
PROBLEM 1. CHIP EXCHANGE

Not submitted yet

English (en)

Bessie the cow has in her possession A chips of type A and B chips of type B ($0 \le A, B \le 10^9$). She can perform the following operation as many times as she likes:

- If you have at least c_B chips of type B, exchange c_B chips of type B for c_A chips of type A ($1 \le c_A, c_B \le 10^9$).

Determine the minimum non-negative integer x such that the following holds: after receiving x additional random chips, it is guaranteed that Bessie can end up with at least f_A chips of type A ($0 \le f_A \le 10^9$).

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains T, the number of independent test cases ($1 \le T \le 10^4$).

Then follow T tests, each consisting of five integers A, B, c_A, c_B, f_A.

OUTPUT FORMAT (print output to the terminal / stdout):

Output the answer for each test on a separate line.

Note: The large size of integers involved in this problem may require the use of 64-bit integer data types (e.g., a "long long" in C/C++).

SAMPLE INPUT:

```
2
2 3 1 1 6
2 3 1 1 4
```

SAMPLE OUTPUT:

```
1
0
```

SAMPLE INPUT:

```
5
0 0 2 3 5
0 1 2 3 5
1 0 2 3 5
10 10 2 3 5
0 0 1 1000000000 1000000000
```

SAMPLE OUTPUT:

```
9
8
7
0
1000000000000000000
```

For the first test, Bessie initially starts with no chips. If she receives any 9 additional chips, she can perform the operation to end up with at least 5 chips of type A. For example, if she receives 2 chips of type A and 7 chips of type B, she can perform the operation twice to end up with $6 \ge 5$ chips of type A. However, if she only receive 8 chips of type B, she can only end up with $4 < 5$ chips of type A.

For the fourth test, she already has enough chips of type A from the start.

SCORING:

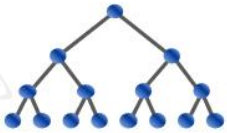
- Input 3: $c_A = c_B = 1$
- Inputs 4-5: $x \le 10$ for all cases
- Inputs 6-7: $c_A = 2$, $c_B = 3$
- Inputs 8-12: No additional constraints.

Problem credits: Benjamin Qi

Language: C

Source File: 选择文件 未选择文件

Submit Solution



USACO 2026 FIRST CONTEST, BRONZE
PROBLEM 2. COW SPLITS

Not submitted yet

English (en)

Bessie is given a positive integer N and a string S of length $3N$ which is generated by concatenating N strings of length 3, each of which is a cyclic shift of "COW". In other words, each string will be "COW", "OWC", or "WCO".

String X is a *square string* if and only if there exists a string Y such that $X = Y + Y$ where $+$ represents string concatenation. For example, "COWCOW" and "CC" are examples of square strings but "COWO" and "OC" are not.

In a single operation, Bessie can remove any **subsequence** T from S where T is a square string. A subsequence of a string is a string which can be obtained by removing several (possibly zero) characters from the original string.

Your job is to help Bessie determine whether it is possible to transform S into an empty string. Additionally, if it is possible, then you must provide a way to do so.

Bessie is also given a parameter k which is either 0 or 1. Let M be the number of operations in your construction.

- If $k = 0$, then M must equal the minimum possible number of operations.
- If $k = 1$, then M can be up to one plus the minimum possible number of operations

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains T, the number of independent test cases ($1 \le T \le 10^4$) and k ($0 \le k \le 1$).

The first line of each test case has N ($1 \le N \le 10^5$).

The second line of each test case has S.

The sum of N across all test cases will not exceed 10^5.

OUTPUT FORMAT (print output to the terminal / stdout):

For each test case, output either one or two lines using the following procedure.

If it is impossible to transform S into an empty string, print -1 on a single line.

Otherwise, on the first line print M -- the number of operations in your construction. On the second line, print $3N$ space-separated integers. The ith integer x indicates that the ith letter of S was deleted as part of the xth subsequence ($1 \le x \le M$).

SAMPLE INPUT:

```
3 1
3
COWOWCWCO
4
WCOCOWWCOCOW
6
COWCOWOWCOWCOWCOWC
```

SAMPLE OUTPUT:

```
-1
1
1 1 1 1 1 1 1 1 1 1 1 1
3
3 3 2 3 3 2 1 1 1 1 1 1 1 1 1 1 1 1
```

For the last test, the optimal number of operations is two, so any valid construction with either $M = 2$ or $M = 3$ would be accepted.

For $M = 3$, here is a possible construction:

1. In the first operation, remove the last twelve characters. Now we're left with COWCOW.
2. In the second operation, remove the subsequence WW. Now we're left with COCO.
3. In the last operation, remove all remaining characters.

SAMPLE INPUT:

```
3 0
3
COWOWCWCO
4
WCOCOWWCOCOW
6
COWCOWOWCOWCOWCOWC
```

SAMPLE OUTPUT:

USACO 2026 FIRST CONTEST, BRONZE
PROBLEM 3. PHOTOSHOOT

Not submitted yet

English (en)

Farmer John is looking at his cows in a magical field and wants to take pictures of subsets of his cows.

The field can be seen as a $N \times N$ grid ($1 \leq N \leq 500$), with a single stationary cow at each location. Farmer John's camera is capable of taking a picture of any $K \times K$ square that is part of the field ($1 \leq K \leq \min(N, 25)$).

At all times, each cow has a beauty value between 0 and 10^6. The attractiveness index of a picture is the sum of the beauty values of the cows contained in the picture.

The beauty value for every cow starts out as 0, so the attractiveness index of any picture in the beginning is 0.

At Q times ($1 \leq Q \leq 3 \cdot 10^4$), the beauty of a single cow will increase by a positive integer due to eating the magical grass that is planted on Farmer John's field.

Farmer John wants to know the maximum attractiveness index of a picture he can take after each of the Q updates.

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains integers N and K.

The following line contains an integer Q.

Each of the following Q lines contains three integers: r, c, and v, which are the row, column, and new beauty value, respectively ($1 \leq r, c \leq N, 1 \leq v \leq 10^6$). It is guaranteed that the new beauty value is greater than the beauty value at that location before.

OUTPUT FORMAT (print output to the terminal / stdout):

Output Q lines, corresponding to the maximum attractiveness index of a picture after each update.

SAMPLE INPUT:

```
4 2
3
2 2 11
3 4 3
3 1 100
```

SAMPLE OUTPUT:

```
11
11
111
```

After the first update, a picture with the maximum attractiveness index is the picture with upper left corner $(2, 2)$ and lower right corner $(3, 3)$, which has an attractiveness index of $11 + 0 + 0 + 0 = 11$.

The second update does not affect the maximum attractiveness index.

After the third update, the picture with the maximum attractiveness index changes to the picture with upper left corner $(2, 1)$ and lower right corner $(3, 2)$, which has an attractiveness index of $0 + 11 + 100 + 0 = 111$.

SAMPLE INPUT:

```
3 1
3
2 2 3
2 2 5
2 2 7
```

SAMPLE OUTPUT:

```
3
5
7
```

There is only one cow with a positive beauty value, so the maximum attractiveness index will always include that cow.

SCORING:

- Inputs 3-6: $N \leq 50, Q \leq 100$
- Inputs 7-10: $N \leq 50$
- Inputs 11-18: No additional constraints.

Problem credits: Brian Law and Cici Liu