A. Madhav's Mistake

1 s., 256 MB

Madhav was once bored in his room after a hectic test and quiz week. The internet was down, so he had nothing to do. So he randomly went to Subham's room, his room was unlocked but he was not in his room at that time. So he decided to wait till he come back. He saw some binary string[†] s (possibly of length 0) on the notebook on his table. Being bored as hell, while waiting he decides to perform the following operation to each string several (possibly zero) times randomly:

- Add 0 to one end of the string and 1 to the other end of the string. For example, starting from the string 1010111, you can obtain either 01010111 or 11010110.

When Subham came back, he saw all his strings have been altered and he was devastated. Madhav feeling guilty asked his if he could do anything to rectify his mistake. Subham said No, but he told if he gave him length of shortest possible string he could have started with, he might be able to restore those strings. Madhav asked you to help him with this, as he's sleepy now and have to study for the test tomorrow. What is the length of the **shortest** possible string he could have started with?

[†] A binary string is a string (possibly the empty string) whose characters are either 0 or 1.

Input

The first line of the input contains an integer t ($1 \le t \le 100$) — the number of testcases.

The first line of each test case contains an integer n ($1 \le n \le 2000$) — the length of final string.

The second line of each test case contains a string s of length n consisting of characters 0 or 1, denoting the final string.

Output

For each test case, output a single nonnegative integer — the shortest possible length of Subham's original string. Note that Subham's original string could have been empty, in which case you should output 0.

input

```
4
3
110
4
1000
3
101
6
110100
```

output

```
1
2
3
0
```

In the first test case, the shortest possible string Subham, had is 1, and Madhav performed the following operation: 1 → 110.

In the second test case, the shortest possible string Subham had is 00, and Madhav performed the following operation: 00 → 1000.

In the third test case, the shortest possible string Subham had is 101, and Madhav didn't perform any operations.

In the fourth test case, the shortest possible string Subahm had is the empty string (which we denote by ε), and Madhav performed the following operations: $\varepsilon \to 01 \to 1010 \to 110100$.

B. Disco Floor

2.0 s, 256 MB

You are given a disco floor represented as a grid with N rows and M columns. Each tile on the disco floor can be colored Red (R), Green (G), or Blue (B). The disco floor is indexed from $(1, 1)$ at the top-left corner to (N, M) at the bottom-right corner.

You need to answer Q queries, where each query is defined by four integers r_1, c_1, r_2, and c_2. These integers represent the coordinates of the top-left corner (r_1, c_1) and bottom-right corner (r_2, c_2) of a subrectangle on the disco floor.

For each query, you are required to output three integers r, g, and b, denoting the count of Red, Green, and Blue tiles, respectively, within the specified subrectangle.

Input

The first line contains three integers N, M, and Q $(1 \le N, M \le 1000, 1 \le Q \le 10^4)$, representing the number of rows, columns, and queries.

The next N lines contain strings of length M, each representing a row of the disco floor. The characters in the string can be either 'R', 'G', or 'B'.

The next Q lines contain four integers each, r_1, c_1, r_2, and c_2 $(1 \le r_1 \le r_2 \le N, 1 \le c_1 \le c_2 \le M)$ denoting the top-left and bottom-right corners of the subrectangles for the queries.

Output

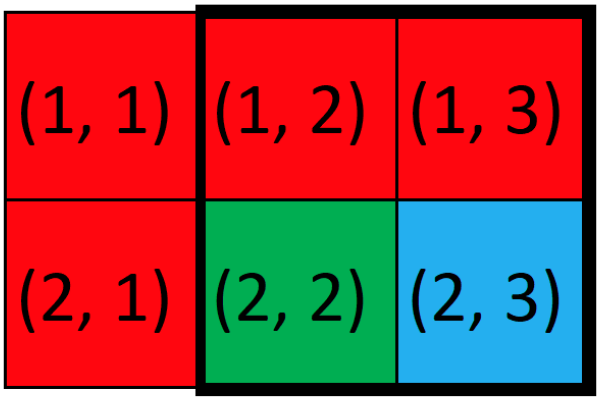
For each of the Q queries, output a line containing three space-separated integers r, g, and b, representing the count of Red, Green, and Blue tiles within the specified subrectangle.

input

```
2 3 5
RRR
RGB
1 2 2 3
2 3 2 3
1 1 2 3
1 1 1 3
2 1 2 3
```

output

```
2 1 1
0 0 1
4 1 1
3 0 0
1 1 1
```



For test 1: For query "1 2 2 3" output should be "2 1 1" as $(1, 2), (1, 3)$ are Red, $(2, 2)$ is Green and $(2, 3)$ is Blue.

C. Yatharth's Superstition

1 s., 256 MB

Yatharth's a workaholic and he dosen't like to sit idle, so once he had no work after a series of tests.

At his room, Yatharth randomly found two strings, a and b, of lengths n and m respectively writen on paper at his roommate's desk. They consist of lowercase English letters and **no character is contained in both strings**.

Being bored, he starts to make another string c, initially empty. Yatharth does the following two types of operations:

- Choose any character from the string a, remove it from a, and add it to the end of c.
- Choose any character from the string b, remove it from b, and add it to the end of c.

Being superstitious, he will not do more than k operations of the same type in a row. He performs operations until either a or b becomes empty. He wants the lexicographically smallest possible value of c after he finishes.

But he realises he hasn't slept for last 48 hours due to all-nighters from last tests, so he leaves to sleep. Since, he is superstitious, he believes he will get a NC if he didn't solve the problem. So he calls you to solve the problem he left. Now, it's on you to solve the problem

A string x is lexicographically smaller than a string y if and only if one of the following holds:

- x is a prefix of y, but $x \neq y$;
- in the first position where x and y differ, the string x has a letter that appears earlier in the alphabet than the corresponding letter in y.

Input
There are several test cases in the input data. The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases. This is followed by the test cases description.

The first line of each test case contains three integers n, m, and k ($1 \leq n, m, k \leq 100$) — parameters from the statement.

The second line of each test case contains the string a of length n.

The third line of each test case contains the string b of length m.

The strings contain only lowercase English letters. It is guaranteed that no symbol appears in a and b simultaneously.

Output
In each test case, output a single string c — the answer to the problem.

input

```
3
8 5 2
aaaaaaaa
bbbbb
15 5 3
befdehjdekdedbp
caaca
7 7 1
wxhtzdy
noskill
```

output

```
aabaabaabaa
aaabbcc
dihktlwlxnyoz
```

In the first test case, it is optimal to take two 'a's from the string a and add them to the string c. Then it is forbidden to take more characters from a, hence one character 'b' from the string b has to be taken. Following that logic, we end up with c being 'aabaabaabaa' when string a is emptied.

In the second test case it is optimal to take as many 'a's from string a as possible, then take as many 'b's as possible from string b. In the end, we take two 'c's from the string a emptying it.

D. Dua Lipa's Next Hit

2 s., 256 MB

After her latest release "Houdini", Dua Lipa is already working on her next song. She has listed an array of musical notes, each with its own melodic frequency.

You are given an array $s_1, s_2, \ldots, s_n$, where all elements are different.

You are her producer and lead songwriter, your task is to carry out a series of "balancing operations" on this array to maximise the melodic sum. You have to do **exactly** x "balancing operations"; during each "balancing operation", you have to do **exactly one** of the following two actions (you choose which operation to do yourself):

- Eliminate the **two** notes that have the **least** melodic frequency; OR
- Eliminate the note that has the **maximum** melodic frequency.

You have to output the maximum possible sum of elements in the resulting array.

Input
The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases.

Each test case consists of two lines:

- the first line contains two integers n and x ($3 \leq n \leq 2 \cdot 10^5$; $1 \leq x \leq 99999$; $2x < n$) — the number of elements and operations, respectively.
- the second line contains n integers $s_1, s_2, \ldots, s_n$ ($1 \leq s_i \leq 10^9$; all s_i are different) — the elements of the array.

Additional constraint on the input: the sum of n does not exceed $2 \cdot 10^5$.

Output
For each test case, print one integer — the maximum possible sum of elements in the resulting array.

input

```
6
5 1
3 5 2 10 6
5 2
2 5 1 10 6
3 1
1 2 3
6 1
15 22 12 10 13 11
6 2
15 22 12 10 13 11
5 1
999999996 999999999 999999997 999999998 999999995
```

output

```
21
11
3
62
46
3999999986
```

In the first testcase, applying the first operation produces the following outcome:

- two minimums are 2 and 3; removing them leaves the array as $[5, 10, 6]$, with sum 21;
- a maximum is 10; removing it leaves the array as $[3, 5, 2, 6]$, with sum 16.

21 is the best answer.

In the second testcase, it's optimal to first erase two minimums, then a maximum.

E. Ritvik Indulges

1 second, 256 megabytes

Your best friend Ritvik has recently learnt about the world of hacking and cybersecurity. To help him indulge in his new hobby, you present him with an encrypted bit string.

The encrypted bit string is represented by a sequence of $0s$ and $1s$ (denoted by s), of size N. He is allowed to perform one of two operations at a time, each taking precisely **one second** to process:

Select Operation: Ritvik can **select** a bit in the string.

Shift Operation: Ritvik can shift the currently selected bit one index to the right (i.e., exchange s_i and s_{i+1}, but only if the next bit is not 1. He can repeat the shift operation as many times as he wants, provided he has selected a bit.

The decryption process is considered successful when all the $1s$ in the string occupy the rightmost bits. Ritvik is determined to enjoy his role-playing as a "hacker" for **as long as possible**.

Determine the **maximum** time he can indulge in his role-play as a "hacker".

Input
The first line of the input contains an integer T denoting the number of test cases. $(1 \leq T \leq 10)$.

The only line of each test case contains the string s. $(1 \leq N \leq 10^5)$.

Output
For each test case, output a single line containing one integer representing the maximum time Ritvik can role-play as a "hacker".

input
```
4
1000
00000111
100101
0
```
output
```
4
0
7
0
```

test 1: He can select the first bit (s_0) and shift it three times, which would take $1 + 3 = 4$ seconds.

test 2: the string is already decrypted, so answer is 0 seconds.

test 3: He first selects the first bit (s_0) and shifts it two times. s would become 001101 after 3 seconds. He then selects s_3 and shifts it one time, which further takes 2 seconds to get 001011. Now, he again selects s_2 and shifts it one time to decrypt the string. It takes $3 + 2 + 2 = 7$ seconds to decrypt the string.

test 4: string is already decrypted as there are no $1s$ in the string.

F. Master Hacker Ritvik

1 s., 512 MB

Ritvik has spent some time working in cybersecurity, and now has gained a worldwide reputation as a master hacker. However, this comes with its downsides - someone is now trying to hack Ritvik's computer and make it crash!

After analysing the traffic, Ritvik realises that the hacker is trying to disrupt his computer by reordering packets coming over the network. To better understand the hack, he sends n packets of data to himself, numbered from 1 to n. Hence, he knows both the order in which he sent the data packets and the order in which he received them.

Ritvik has noticed that the number and IDs of packets sent and received are always the same, however, they may be received in a different order than he sent them. A packet i is said to be *out of order* if it was received before a packet j that it was sent after and must be discarded. If he is unsure, Ritvik prefers to assume a packet was not out of order.

Find the minimum number of packets that were out of order.

Input
The first line contains a single integer n $(2 \leq n \leq 10^5)$, denoting the number of data packets Ritvik sent.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq n)$, denoting IDs of the packets in the order they were sent.

The third line contains n integers $b_1, b_2, \ldots, b_n$ $(1 \leq b_i \leq n)$, denoting IDs of the packets in the order they were received.

Output
Output p, the *minimum* number of packets that must have been out of order.

input
```
2
2 1
2 1
```
output
```
0
```

input
```
3
1 2 3
3 1 2
```
output
```
1
```

In testcase 1, no packet is received out of order.

In testcase 2, packet 3 is received out of order.

G. Homecoming Feast

2 s., 256 MB

Yatharth loves his mother's food and has just arrived back at his home after the compres. To celebrate his homecoming, Yatharth's mother has prepared a feast for him. There are N dishes placed in a row. The calorie content of the i-th dish is a_i, and no two dishes have the same number of calories.

Yatharth devours food in two styles: he either spends x minutes eating k consecutively placed dishes; or he spends y minutes, picks two consecutively placed dishes and eats the one with lower calorie content.

Yatharth's stomach, however, has limited capacity. He eats some dishes and has to leave some. You have a list of the calories of the dishes remaining, in the same order, given by b_i.

For example, let the calories of the dishes be $[2, 3, 7, 8, 11, 5, 4]$, and $k = 3$. If Yatharth picks up dishes with calories 8 and 11, the remaining dishes becomes $[2, 3, 7, 11, 5, 4]$. Then, for example, if he eats consecutive dishes $[7, 11, 5]$, the remaining dishes become $[2, 3, 4]$.

Basically, Yatharth converts the dishes on the table from $a_1, a_2, \ldots, a_n$ to $b_1, b_2, \ldots, b_m$ (using two specific eating styles). Calculate the minimum time (in minutes) Yatharth took relishing the feast.

Input
The first line contains two integers n and m $(1 \leq n, m \leq 2 \cdot 10^5)$ — the number of dishes initially (length of array a) and the number of dishes remaining (length of array b) respectively.

The second line contains three integers x, k, y $(1 \leq x, y, \leq 10^9; 1 \leq k \leq n)$ — the time (in minutes) taken in eating k consecutive dishes, number of consecutive dishes eaten at a time in this style, and the time (in minutes) taken in picking up two dishes and eating the low-calorie one respectively.

The third line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq n)$. Note that all integers a_i are distinct.

The fourth line contains m integers $b_1, b_2, \ldots, b_m$ $(1 \leq b_i \leq n)$. Note that all integers b_i are distinct.

Output

Print the minimum time (in minutes) for turning array a into array b, or -1 if it is impossible.

input

```
7 3
5 3 2
2 3 6 1 7 5 4
3 6 4
```

output

```
7
```

input

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

output

```
-1
```