Educational Codeforces Round 120 (Rated for Div. 2)

A. Construct a Rectangle

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are three sticks with integer lengths l_1, l_2 and l_3.

You are asked to break exactly one of them into two pieces in such a way that:

- both pieces have positive (strictly greater than 0) **integer** length;
- the total length of the pieces is equal to the original length of the stick;
- it's possible to construct a rectangle from the resulting four sticks such that each stick is used as exactly one of its sides.

A square is also considered a rectangle.

Determine if it's possible to do that.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases.

The only line of each testcase contains three integers l_1, l_2, l_3 ($1 \le l_i \le 10^8$) — the lengths of the sticks.

Output

For each testcase, print "YES" if it's possible to break one of the sticks into two pieces with positive integer length in such a way that it's possible to construct a rectangle from the resulting four sticks. Otherwise, print "NO".

You may print every letter in any case you want (so, for example, the strings yEs, yes, Yes and YES are all recognized as a positive answer).

Example

input
4 6 1 5 2 5 2 2 4 2 5 5 4
output
YES NO YES YES

Note

In the first testcase, the first stick can be broken into parts of length 1 and 5. We can construct a rectangle with opposite sides of length 1 and 5.

In the second testcase, breaking the stick of length 2 can only result in sticks of lengths $1, 1, 2, 5$, which can't be made into a rectangle. Breaking the stick of length 5 can produce results $2, 3$ or $1, 4$ but neither of them can't be put into a rectangle.

In the third testcase, the second stick can be broken into parts of length 2 and 2. The resulting rectangle has opposite sides 2 and 2 (which is a square).

In the fourth testcase, the third stick can be broken into parts of length 2 and 2. The resulting rectangle has opposite sides 2 and 5.

B. Berland Music

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Berland Music is a music streaming service built specifically to support Berland local artist. Its developers are currently working on a song recommendation module.

So imagine Monocarp got recommended n songs, numbered from 1 to n. The i-th song had its predicted rating equal to p_i, where $1 \le p_i \le n$ and every integer from 1 to n appears exactly once. In other words, p is a permutation.

After listening to each of them, Monocarp pressed either a like or a dislike button. Let his vote sequence be represented with a string s, such that $s_i = 0$ means that he disliked the i-th song, and $s_i = 1$ means that he liked it.

Now the service has to re-evaluate the song ratings in such a way that:

- the new ratings $q_1, q_2, \ldots, q_n$ still form a permutation ($1 \le q_i \le n$; each integer from 1 to n appears exactly once);
- every song that Monocarp liked should have a greater rating than every song that Monocarp disliked (formally, for all i, j such that $s_i = 1$ and $s_j = 0$, $q_i > q_j$ should hold).

Among all valid permutations q find the one that has the smallest value of $\sum\limits_{i=1}^{n} |p_i - q_i|$, where $|x|$ is an absolute value of x.

Print the permutation $q_1, q_2, \ldots, q_n$. If there are multiple answers, you can print any of them.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases.

The first line of each testcase contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of songs.

The second line of each testcase contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$) — the permutation of the predicted ratings.

The third line contains a single string s, consisting of n characters. Each character is either a 0 or a 1. 0 means that Monocarp disliked the song, and 1 means that he liked it.

The sum of n over all testcases doesn't exceed $2 \cdot 10^5$.

Output

For each testcase, print a permutation q — the re-evaluated ratings of the songs. If there are multiple answers such that $\sum\limits_{i=1}^{n} |p_i - q_i|$ is minimum possible, you can print any of them.

Example

input
3 2 1 2 10 3 3 1 2 111 8 2 3 1 8 5 4 7 6 01110001
output
2 1 3 1 2 1 6 5 8 3 2 4 7

Note

In the first testcase, there exists only one permutation q such that each liked song is rating higher than each disliked song: song 1 gets rating 2 and song 2 gets rating 1. $\sum\limits_{i=1}^{n} |p_i - q_i| = |1 - 2| + |2 - 1| = 2$.

In the second testcase, Monocarp liked all songs, so all permutations could work. The permutation with the minimum sum of absolute differences is the permutation equal to p. Its cost is 0.

C. Set or Decrease

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an integer array $a_1, a_2, \ldots, a_n$ and integer k.

In one step you can

- either choose some index i and decrease a_i by one (make $a_i = a_i - 1$);
- or choose two indices i and j and set a_i equal to a_j (make $a_i = a_j$).

What is the minimum number of steps you need to make the sum of array $\sum\limits_{i=1}^{n} a_i \le k$? (You are allowed to make values of array negative).

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n and k ($1 \le n \le 2 \cdot 10^5$; $1 \le k \le 10^{15}$) — the size of array a and upper bound on its sum.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the array itself.

It's guaranteed that the sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print one integer — the minimum number of steps to make $\sum_{i=1}^{n} a_i \le k$.

Example

input
4 1 10 20 2 69 6 9 7 8 1 2 1 3 1 2 1 10 1 1 2 3 1 2 6 1 6 8 10
output
10 0 2 7

Note

In the first test case, you should decrease a_1 10 times to get the sum lower or equal to $k = 10$.

In the second test case, the sum of array a is already less or equal to 69, so you don't need to change it.

In the third test case, you can, for example:

1. set $a_4 = a_3 = 1$;
2. decrease a_4 by one, and get $a_4 = 0$.

As a result, you'll get array $[1, 2, 1, 0, 1, 2, 1]$ with sum less or equal to 8 in $1 + 1 = 2$ steps.
In the fourth test case, you can, for example:

1. choose a_7 and decrease in by one 3 times; you'll get $a_7 = -2$;
2. choose 4 elements a_6, a_8, a_9 and a_{10} and them equal to $a_7 = -2$.

As a result, you'll get array $[1, 2, 3, 1, 2, -2, -2, -2, -2, -2]$ with sum less or equal to 1 in $3 + 4 = 7$ steps.

D. Shuffle

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given a binary string (i. e. a string consisting of characters `0` and/or `1`) s of length n. You can perform the following operation with the string s **at most once**: choose a substring (a contiguous subsequence) of s having **exactly** k characters `1` in it, and shuffle it (reorder the characters in the substring as you wish).

Calculate the number of different strings which can be obtained from s by performing this operation at most once.

Input

The first line contains two integers n and k ($2 \le n \le 5000$; $0 \le k \le n$).

The second line contains the string s of length n, consisting of characters `0` and/or `1`.

Output

Print one integer — the number of different strings which can be obtained from s by performing the described operation at most once. Since the answer can be large, output it modulo 998244353.

Examples

input
7 2 1100110
output
16

input

5 0 10010
output
1

input
8 1 10001000
output
10

input
10 8 0010011000
output
1

Note

Some strings you can obtain in the first example:

- to obtain `0110110`, you can take the substring from the 1-st character to the 4-th character, which is `1100`, and reorder its characters to get `0110`;
- to obtain `1111000`, you can take the substring from the 3-rd character to the 7-th character, which is `00110`, and reorder its characters to get `11000`;
- to obtain `1100101`, you can take the substring from the 5-th character to the 7-th character, which is `110`, and reorder its characters to get `101`.

In the second example, $k = 0$ so you can only choose the substrings consisting only of `0` characters. Reordering them doesn't change the string at all, so the only string you can obtain is `10010`.

E. Math Test

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya is a math teacher. n of his students has written a test consisting of m questions. For each student, it is known which questions he has answered correctly and which he has not.

If the student answers the j-th question correctly, he gets p_j points (otherwise, he gets 0 points). Moreover, the points for the questions are distributed in such a way that the array p is a permutation of numbers from 1 to m.

For the i-th student, Petya knows that he expects to get x_i points for the test. Petya wonders how unexpected the results could be. Petya believes that the *surprise value* of the results for students is equal to $\sum\limits_{i=1}^{n} |x_i - r_i|$, where r_i is the number of points that the i-th student has got for the test.

Your task is to help Petya find such a permutation p for which the *surprise value* of the results is maximum possible. If there are multiple answers, print any of them.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n and m ($1 \le n \le 10$; $1 \le m \le 10^4$) — the number of students and the number of questions, respectively.

The second line contains n integers $x_1, x_2, \ldots, x_n$ ($0 \le x_i \le \frac{m(m+1)}{2}$), where x_i is the number of points that the i-th student expects to get.

This is followed by n lines, the i-th line contains the string s_i ($|s_i| = m; s_{i,j} \in \{0, 1\}$), where $s_{i,j}$ is 1 if the i-th student has answered the j-th question correctly, and 0 otherwise.

The sum of m for all test cases does not exceed 10^4.

Output

For each test case, print m integers — a permutation p for which the *surprise value* of the results is maximum possible. If there are multiple answers, print any of them.

Example

input
3

4 3 5 1 2 2 110 100 101 100 4 4 6 2 0 10 1001 0010 0110 0101 3 6 20 3 15 010110 000101 111111
output
3 1 2 2 3 4 1 3 1 4 5 2 6

F. Quadratic Set

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let's call a set of positive integers $a_1, a_2, \ldots, a_k$ *quadratic* if the product of the factorials of its elements is a square of an integer, i.e. $\prod_{i=1}^{k} a_i! = m^2$, for some integer m.

You are given a positive integer n.

Your task is to find a *quadratic* subset of a set $1, 2, \ldots, n$ of maximum size. If there are multiple answers, print any of them.

Input

A single line contains a single integer n ($1 \leq n \leq 10^6$).

Output

In the first line, print a single integer — the size of the maximum subset. In the second line, print the subset itself in an arbitrary order.

Examples

input
1
output
1 1

input
4
output
3 1 3 4

input
7
output
4 1 4 5 6

input
9
output
7 1 2 4 5 6 7 9