A. Tokitsukaze and All Zero Sequence

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Tokitsukaze has a sequence a of length n. For each operation, she selects two numbers a_i and a_j ($i \neq j$; $1 \leq i, j \leq n$).

- If $a_i = a_j$, change one of them to 0.
- Otherwise change both of them to $\min(a_i, a_j)$.

Tokitsukaze wants to know the minimum number of operations to change all numbers in the sequence to 0. It can be proved that the answer always exists.

Input

The first line contains a single positive integer t ($1 \leq t \leq 1000$) — the number of test cases.

For each test case, the first line contains a single integer n ($2 \leq n \leq 100$) — the length of the sequence a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 100$) — the sequence a.

Output

For each test case, print a single integer — the minimum number of operations to change all numbers in the sequence to 0.

Example

input

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

output

```
4
3
2
```

Note

In the first test case, one of the possible ways to change all numbers in the sequence to 0:

In the 1-st operation, $a_1 < a_2$, after the operation, $a_2 = a_1 = 1$. Now the sequence a is $[1, 1, 3]$.

In the 2-nd operation, $a_1 = a_2 = 1$, after the operation, $a_1 = 0$. Now the sequence a is $[0, 1, 3]$.

In the 3-rd operation, $a_1 < a_2$, after the operation, $a_2 = 0$. Now the sequence a is $[0, 0, 3]$.

In the 4-th operation, $a_2 < a_3$, after the operation, $a_3 = 0$. Now the sequence a is $[0, 0, 0]$.

So the minimum number of operations is 4.

B1. Tokitsukaze and Good 01-String (easy version)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is the easy version of the problem. The only difference between the two versions is that the harder version asks additionally for a minimum number of subsegments.

Tokitsukaze has a binary string s of length n, consisting only of zeros and ones, n is **even**.

Now Tokitsukaze divides s into **the minimum number** of **contiguous** subsegments, and for each subsegment, all bits in each subsegment are the same. After that, s is considered good if the lengths of all subsegments are even.

For example, if s is "`11001111`", it will be divided into "`11`", "`00`" and "`1111`". Their lengths are 2, 2, 4 respectively, which are all even numbers, so "`11001111`" is good. Another example, if s is "`1110011000`", it will be divided into "`111`", "`00`", "`11`" and "`000`", and their lengths are 3, 2, 2, 3. Obviously, "`1110011000`" is not good.

Tokitsukaze wants to make s good by changing the values of some positions in s. Specifically, she can perform the operation any number of times: change the value of s_i to '0' or '1'($1 \leq i \leq n$). Can you tell her the minimum number of operations to make s good?

Input

The first contains a single positive integer t ($1 \leq t \leq 10\,000$) — the number of test cases.

For each test case, the first line contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$) — the length of s, it is guaranteed that n is even.

The second line contains a binary string s of length n, consisting only of zeros and ones.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single line with one integer — the minimum number of operations to make s good.

Example

input

```
5
10
1110011000
8
11001111
2
00
2
11
6
100110
```

output

```
3
0
0
0
3
```

Note

In the first test case, one of the ways to make s good is the following.

Change s_3, s_6 and s_7 to '0', after that s becomes "1100000000", it can be divided into "11" and "00000000", which lengths are 2 and 8 respectively. There are other ways to operate 3 times to make s good, such as "1111110000", "1100001100", "1111001100".

In the second, third and fourth test cases, s is good initially, so no operation is required.

B2. Tokitsukaze and Good 01-String (hard version)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is the hard version of the problem. The only difference between the two versions is that the harder version asks additionally for a minimum number of subsegments.

Tokitsukaze has a binary string s of length n, consisting only of zeros and ones, n is **even**.

Now Tokitsukaze divides s into **the minimum number** of **contiguous** subsegments, and for each subsegment, all bits in each subsegment are the same. After that, s is considered good if the lengths of all subsegments are even.

For example, if s is "11001111", it will be divided into "11", "00" and "1111". Their lengths are 2, 2, 4 respectively, which are all even numbers, so "11001111" is good. Another example, if s is "1110011000", it will be divided into "111", "00", "11" and "000", and their lengths are 3, 2, 2, 3. Obviously, "1110011000" is not good.

Tokitsukaze wants to make s good by changing the values of some positions in s. Specifically, she can perform the operation any number of times: change the value of s_i to '0' or '1' ($1 \leq i \leq n$). Can you tell her the minimum number of operations to make s good? **Meanwhile, she also wants to know the minimum number of subsegments that s can be divided into among all solutions with the minimum number of operations.**

Input

The first contains a single positive integer t ($1 \leq t \leq 10\,000$) — the number of test cases.

For each test case, the first line contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$) — the length of s, it is guaranteed that n is even.

The second line contains a binary string s of length n, consisting only of zeros and ones.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single line with two integers — the minimum number of operations to make s good, and the minimum number of subsegments that s can be divided into among all solutions with the minimum number of operations.

Example

input

```
5
10
1110011000
8
11001111
2
00
2
11
6
100110
```

output

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

Note

In the first test case, one of the ways to make s good is the following.

Change s_3, s_6 and s_7 to '0', after that s becomes "1100000000", it can be divided into "11" and "00000000", which lengths are 2 and 8 respectively, the number of subsegments of it is 2. There are other ways to operate 3 times to make s good, such as "1111110000", "1100001100", "1111001100", the number of subsegments of them are 2, 4, 4 respectively. It's easy to find that the minimum number of subsegments among all solutions with the minimum number of operations is 2.

In the second, third and fourth test cases, s is good initially, so no operation is required.

C. Tokitsukaze and Strange Inequality

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Tokitsukaze has a permutation p of length n. Recall that a permutation p of length n is a sequence $p_1, p_2, \ldots, p_n$ consisting of n distinct integers, each of which from 1 to n ($1 \leq p_i \leq n$).

She wants to know how many different indices tuples $[a, b, c, d]$ ($1 \leq a < b < c < d \leq n$) in this permutation satisfy the following two inequalities:

$$p_a < p_c \text{ and } p_b > p_d.$$

Note that two tuples $[a_1, b_1, c_1, d_1]$ and $[a_2, b_2, c_2, d_2]$ are considered to be different if $a_1 \neq a_2$ or $b_1 \neq b_2$ or $c_1 \neq c_2$ or $d_1 \neq d_2$.

Input

The first line contains one integer t ($1 \leq t \leq 1000$) — the number of test cases. Each test case consists of two lines.

The first line contains a single integer n ($4 \leq n \leq 5000$) — the length of permutation p.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \leq p_i \leq n$) — the permutation p.

It is guaranteed that the sum of n over all test cases does not exceed 5000.

Output

For each test case, print a single integer — the number of different $[a, b, c, d]$ tuples.

Example

input

```
3
6
5 3 6 1 4 2
4
1 2 3 4
10
5 1 6 2 8 3 4 10 9 7
```

output

```
3
0
28
```

Note

In the first test case, there are 3 different $[a, b, c, d]$ tuples.

$p_1 = 5$, $p_2 = 3$, $p_3 = 6$, $p_4 = 1$, where $p_1 < p_3$ and $p_2 > p_4$ satisfies the inequality, so one of $[a, b, c, d]$ tuples is $[1, 2, 3, 4]$.

Similarly, other two tuples are $[1, 2, 3, 6]$, $[2, 3, 5, 6]$.

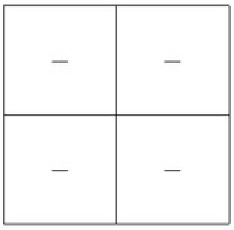
D. Tokitsukaze and Meeting

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

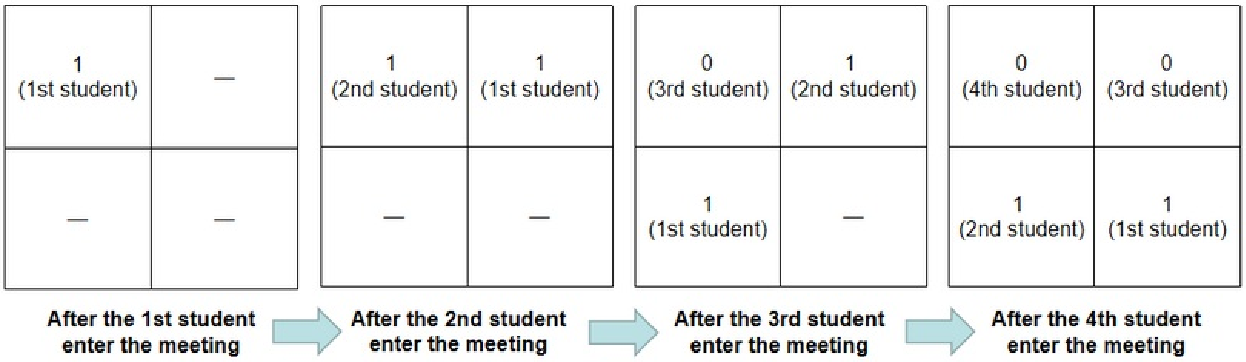
Tokitsukaze is arranging a meeting. There are n rows and m columns of seats in the meeting hall.

There are exactly $n \cdot m$ students attending the meeting, including several naughty students and several serious students. The students are numerated from 1 to $n \cdot m$. The students will enter the meeting hall in order. When the i-th student enters the meeting hall, he will sit in the 1-st column of the 1-st row, and the students who are already seated will move back one seat. Specifically, the student sitting in the j-th ($1 \le j \le m-1$) column of the i-th row will move to the $(j+1)$-th column of the i-th row, and the student sitting in m-th column of the i-th row will move to the 1-st column of the $(i+1)$-th row.

For example, there is a meeting hall with 2 rows and 2 columns of seats shown as below:



There will be 4 students entering the meeting hall in order, represented as a binary string "`1100`", of which '`0`' represents naughty students and '`1`' represents serious students. The changes of seats in the meeting hall are as follows:



Denote a row or a column good if and only if there is at least one serious student in this row or column. Please predict the number of good rows and columns just after the i-th student enters the meeting hall, for all i.

Input

The first contains a single positive integer t ($1 \le t \le 10\,000$) — the number of test cases.

For each test case, the first line contains two integers n, m ($1 \le n, m \le 10^6$; $1 \le n \cdot m \le 10^6$), denoting there are n rows and m columns of seats in the meeting hall.

The second line contains a binary string s of length $n \cdot m$, consisting only of zeros and ones. If s_i equal to '`0`' represents the i-th student is a naughty student, and s_i equal to '`1`' represents the i-th student is a serious student.

It is guaranteed that the sum of $n \cdot m$ over all test cases does not exceed 10^6.

Output

For each test case, print a single line with $n \cdot m$ integers — the number of good rows and columns just after the i-th student enters the meeting hall.

Example

input

```
3
2 2
1100
4 2
11001101
2 4
11001101
```

output

```
2 3 4 3
2 3 4 3 5 4 6 5
2 3 3 3 4 4 4 5
```

Note

The first test case is shown in the statement.

After the 1-st student enters the meeting hall, there are 2 good rows and columns: the 1-st row and the 1-st column.

After the 2-nd student enters the meeting hall, there are 3 good rows and columns: the 1-st row, the 1-st column and the 2-nd column.

After the 3-rd student enters the meeting hall, the 4 rows and columns are all good.

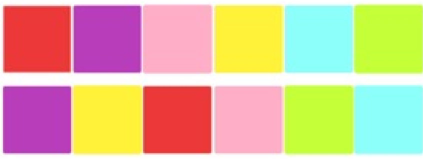
After the 4-th student enters the meeting hall, there are 3 good rows and columns: the 2-nd row, the 1-st column and the 2-nd column.

E. Tokitsukaze and Two Colorful Tapes

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

Tokitsukaze has two colorful tapes. There are n distinct colors, numbered 1 through n, and each color appears exactly once on each of the two tapes. Denote the color of the i-th position of the first tape as ca_i, and the color of the i-th position of the second tape as cb_i.

Now Tokitsukaze wants to select each color an integer value from 1 to n, distinct for all the colors. After that she will put down the color values in each colored position on the tapes. Denote the number of the i-th position of the first tape as $numa_i$, and the number of the i-th position of the second tape as $numb_i$.



For example, for the above picture, assuming that the color red has value x ($1 \leq x \leq n$), it appears at the 1-st position of the first tape and the 3-rd position of the second tape, so $numa_1 = numb_3 = x$.

Note that each color i from 1 to n should have a **distinct** value, and the same color which appears in both tapes has the same value.

After labeling each color, the beauty of the two tapes is calculated as

$$\sum_{i=1}^{n} |numa_i - numb_i|.$$

Please help Tokitsukaze to find the highest possible beauty.

Input

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

For each test case, the first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of colors.

The second line contains n integers $ca_1, ca_2, \ldots, ca_n$ ($1 \leq ca_i \leq n$) — the color of each position of the first tape. It is guaranteed that ca is a permutation.

The third line contains n integers $cb_1, cb_2, \ldots, cb_n$ ($1 \leq cb_i \leq n$) — the color of each position of the second tape. It is guaranteed that cb is a permutation.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the highest possible beauty.

Example

input

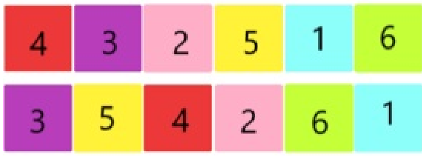
```
3
6
1 5 4 3 2 6
5 3 1 4 6 2
6
3 5 4 6 2 1
3 6 4 5 2 1
1
1
1
```

output

```
18
10
0
```

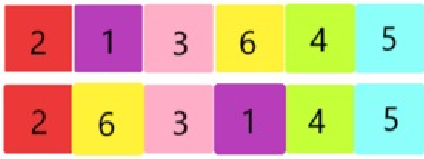
Note

An optimal solution for the first test case is shown in the following figure:



The beauty is $|4-3|+|3-5|+|2-4|+|5-2|+|1-6|+|6-1|=18$.

An optimal solution for the second test case is shown in the following figure:



The beauty is $|2-2|+|1-6|+|3-3|+|6-1|+|4-4|+|5-5|=10$.

F. Tokitsukaze and Permutations

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

Tokitsukaze has a permutation p. She performed the following operation to p **exactly** k times: in one operation, for each i from 1 to $n-1$ in order, if p_i > p_{i+1}, swap p_i, p_{i+1}. After exactly k times of operations, Tokitsukaze got a new sequence a, obviously the sequence a is also a permutation.

After that, Tokitsukaze wrote down the value sequence v of a on paper. Denote the value sequence v of the permutation a of length n as $v_i=\sum_{j=1}^{i-1}[a_i < a_j]$, where the value of $[a_i < a_j]$ define as if $a_i < a_j$, the value is 1, otherwise is 0 (in other words, v_i is equal to the number of elements greater than a_i that are to the left of position i). Then Tokitsukaze went out to work.

There are three naughty cats in Tokitsukaze's house. When she came home, she found the paper with the value sequence v to be bitten out by the cats, leaving several holes, so that the value of some positions could not be seen clearly. She forgot what the original permutation p was. She wants to know how many different permutations p there are, so that the value sequence v of the new permutation a after **exactly** k operations is the same as the v written on the paper (not taking into account the unclear positions).

Since the answer may be too large, print it modulo $998\,244\,353$.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases. Each test case consists of two lines.

The first line contains two integers n and k ($1 \le n \le 10^6$; $0 \le k \le n-1$) — the length of the permutation and the exactly number of operations.

The second line contains n integers $v_1, v_2, \dots, v_n$ ($-1 \le v_i \le i-1$) — the value sequence v. $v_i = -1$ means the i-th position of v can't be seen clearly.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, print a single integer — the number of different permutations modulo $998\,244\,353$.

Example

input

```
3
5 0
0 1 2 3 4
5 2
-1 1 2 0 0
5 2
0 1 1 0 0
```

output

```
1
6
6
```

Note

In the first test case, only permutation $p=[5,4,3,2,1]$ satisfies the constraint condition.

In the second test case, there are 6 permutations satisfying the constraint condition, which are:

- $[3,4,5,2,1] \rightarrow [3,4,2,1,5] \rightarrow [3,2,1,4,5]$
- $[3,5,4,2,1] \rightarrow [3,4,2,1,5] \rightarrow [3,2,1,4,5]$
- $[4,3,5,2,1] \rightarrow [3,4,2,1,5] \rightarrow [3,2,1,4,5]$

- $[4,5,3,2,1] \rightarrow [4,3,2,1,5] \rightarrow [3,2,1,4,5]$
- $[5,3,4,2,1] \rightarrow [3,4,2,1,5] \rightarrow [3,2,1,4,5]$
- $[5,4,3,2,1] \rightarrow [4,3,2,1,5] \rightarrow [3,2,1,4,5]$

So after exactly 2 times of swap they will all become $a = [3,2,1,4,5]$, whose value sequence is $v = [0,1,2,0,0]$.

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform