You are given a binary string $s$ of length $n$.

Let's define $d_i$ as the number whose decimal representation is $s_i s_{i+1}$ (possibly, with a leading zero). We define $f(s)$ to be the sum of all the valid $d_i$. In other words, $f(s) = \sum_{i=1}^{n-1} d_i$.

For example, for the string $s = 1011$:

- $d_1 = 10$ (ten);
- $d_2 = 01$ (one)
- $d_3 = 11$ (eleven);
- $f(s) = 10 + 01 + 11 = 22$.

In one operation you can swap any two adjacent elements of the string. Find the minimum value of $f(s)$ that can be achieved if at most $k$ operations are allowed.

## Input

Each test contains multiple test cases. The first line contains the number of test cases $t$ ($1 \le t \le 10^5$). Description of the test cases follows.

First line of each test case contains two integers $n$ and $k$ ($2 \le n \le 10^5, 0 \le k \le 10^9$) — the length of the string and the maximum number of operations allowed.

The second line of each test case contains the binary string $s$ of length $n$, consisting of only zeros and ones.

It is also given that sum of $n$ over all the test cases doesn't exceed $10^5$.

## Output

For each test case, print the minimum value of $f(s)$ you can obtain with at most $k$ operations.

## Sample 1

| Input copy | Output copy |
|---|---|
| 3<br>4 0<br>1010<br>7 1<br>0010100<br>5 2<br>00110 | 21<br>22<br>12 |

## Note

- For the first example, you can't do any operation so the optimal string is $s$ itself. $f(s) = f(1010) = 10 + 01 + 10 = 21$.
- For the second example, one of the optimal strings you can obtain is "0011000". The string has an $f$ value of 22.
- For the third example, one of the optimal strings you can obtain is "00011". The string has an $f$ value of 12.