UTS Open '24 P3 - LXghts Out

Time limit: 1.0s **Memory limit:** 512M

A new puzzle has begun trending recently: LXghts Out!

The puzzle consists of a row of N lights, each of which is either on or off, represented by `1` and `0`, respectively. In one move, you can choose any light and set its on/off state to the **bitwise xor** of its neighbours (adjacent lights). In other words,

- You can only turn a light **on** if **exactly** 1 of its neighbours is on. $1\underline{0}0 \rightarrow 1\underline{1}0$ or $0\underline{0}1 \rightarrow 0\underline{1}1$
- You can only turn a light **off** if 0 or 2 of its neighbours are on. $0\underline{1}0 \rightarrow 0\underline{0}0$ or $1\underline{1}1 \rightarrow 1\underline{0}1$

Note that light 1 and light N only have 1 neighbour. You can consider the left of light 1 and the right of light N as "off".

The goal of the game is to turn off all of the lights with the minimum number of moves.

All the cool kids in town are racing each other to see who can complete the puzzle the fastest, but you, the programmer, have a few tricks up your sleeve. Before you begin racing, you will use your insane hacking skills to flip the on/off state of at most K different lights in the puzzle.

Can you find the minimum number of moves to solve the puzzle after flipping at most K lights? If the puzzle is impossible regardless of how you flip the lights, output `-1`.

Constraints

$1 \leq N \leq 10^6$
$0 \leq K \leq N$

Subtask 1 [40%]

$K = 0$

Subtask 2 [60%]

$K \geq 0$

Input Specification

The first line of input contains two integers, N and K, the number of lights and the maximum number of flips you can make.

The next line contains N characters, describing the initial state of the lights. `1` represents on and `0` represents off.

Output Specification

Output one line containing one integer, the minimum number of moves to solve the puzzle after flipping at most K different lights.

If the puzzle is always impossible, output `-1` instead.

Sample Input 1

```
6 1
110110
```

Sample Output 1

```
5
```

Explanation for Sample 1

One possible solution is to flip the first light, to get `010110`.

Then, you can solve the puzzle in 5 moves:

`010110`
`000110` - turn off light 2 (0 neighbours on)
`000111` - turn on light 6 (1 neighbour on)
`000101` - turn off light 5 (2 neighbours on)
`000001` - turn off light 4 (0 neighbours on)
`000000` - turn off light 6 (0 neighbours on)

It can be shown that 5 moves is minimal.

Sample Input 2

```
4 2
0000
```

Sample Output 2

```
0
```

Explanation for Sample 2

In this example, it is optimal to not flip any lights beforehand, since the puzzle is already solved! Thus, the minimum number of moves afterwards is 0.

Sample Input 3

```
2 0
11
```

Sample Output 3

```
-1
```

Explanation for Sample 3

You cannot flip any lights beforehand, since $K = 0$. The puzzle is impossible because both lights have exactly 1 neighbour with a light on, and thus, neither of them can be turned off.