CCC '16 S5 - Circle of Life

Time limit: 2.0s **Memory limit:** 256M

Canadian Computing Competition: 2016 Stage 1, Senior #5

You may have heard of *Conway's Game of Life*, which is a simple set of rules for cells on a grid that can produce incredibly complex configurations. In this problem we will deal with a simplified version of the game.

There is a one-dimensional circular strip of N cells. The cells are numbered from 1 to N in the order you would expect: that is, cell 1 and cell 2 are adjacent, cell 2 and cell 3 are adjacent, and so on up to cell $N-1$, which is adjacent to cell N. Since the trip is circular, cell 1 is also adjacent to cell N.

Each cell is either alive (represented by a `1`) or dead (represented by a `0`). The cells change over a number of generations. If **exactly** one of the cell's neighbours is alive in the current generation, then the cell will be alive in the next generation. Otherwise, the cell will be dead in the next generation.

Given the initial state of the strip, find the state after T generations.

Input Specification

The first line will contain two space-separated integers N and T ($3 \leq N \leq 100\,000; 1 \leq T \leq 10^{15}$). The second line will contain a string consisting of exactly N characters, representing the initial configuration of the N cells. Each character in the string will be either `0` or `1`. The initial state of cell I is given by the i^{th} character of the string. The character `1` represents an alive cell and the character `0` represents a dead cell.

- For 1 of the 15 available marks, $N \leq 15$ and $T \leq 15$.
- For an additional 6 of the 15 available marks, $N \leq 15$.
- For an additional 4 of the 15 available marks, $N \leq 4\,000$ and $T \leq 10\,000\,000$.

Note that for full marks, solutions will need to handle 64-bit integers. For example:

- in C/C++, the type `long long` should be used;
- in Java, the type `long` should be used;
- in Pascal, the type `int64` should be used.

Output Specification

Output the string of N characters representing the final state of the cells, in the same format and order as the input.

Sample Input 1

```
7 1
0000001
```

Output for Sample Input 1

```
1000010
```

Explanation for Output for Sample Input 1

Cell 1 and cell $N-1$ are adjacent to cell N, and thus alive after one generation.

Sample Input 2

```
5 3
01011
```

Output for Sample Input 2

```
10100
```

Explanation for Output for Sample Input 2

After one generation, configuration becomes `00011`.

After two generations, configuration becomes `10111`.