

Cloudnet

Filename: cloudnet

The Problem

In a brilliant, but unfortunate, accident, Joel has evolved a self-aware AI called Cloudnet. Unfortunately, Cloudnet has decided that it is superior to its creators and has initiated a mission to destroy all of humanity. Joel had initially equipped Cloudnet with a power switch that could be switched off in case of an emergency. However, Cloudnet thought that this made things too easy for the humans. To make things more complicated, Cloudnet has equipped itself with multiple panels of switches, not all of which follow the normal “up is on, down is off” scheme. In fact, the switches are wired up in such a way that you can only disable the system by going through each switch from right to left order, following these rules:

- Rule A: If a switch’s left and right neighbors match each other, but don’t match the middle switch, then flip all 3 switches.
- Rule B: If all 3 switches match, flip only the middle switch.
- Rule C: If only the right switch matches the middle switch, flip the middle switch.
- Rule D: If only the left switch matches the middle switch, flip the left switch.

Your goal is to save the human race by disabling Cloudnet’s power supply and ending its reign of terror. Each panel consists of only one row of at least three switches. When considering the switches at either end of a row, the opposite “end switch” takes the place of the missing immediate neighbor. Your program must calculate the state of the switches after carrying out this procedure. (Note: This state may NOT consist of all the switches being off, but if you can get the switches to this state, then the entire system powers down.)

The Input

Each input will start with a single integer t ($1 \leq t \leq 100$) on the first line. The number t will denote the number of test cases that follow. Each test case will begin with one integer, s ($3 \leq s \leq 20$), denoting the number of switches in the test case. The next line will contain a row of s characters, either ‘0’ or ‘1’, separated by spaces, representing the initial state of the panel of switches for the test case.

The Output

For each test case, output a single line of binary characters representing the final state of each row of switches (which should disable Cloudnet’s power supply) without separating spaces.

Sample Input

```
2
3
1 1 0
10
1 1 0 1 0 0 1 0 1 0
```

Sample Output

```
100
1010101100
```

Sample Input Explanation

In the first case in the sample input, we start with the following configuration:

110

We first apply the switching rule to the 0 on the right end of the panel. To the right of 0 is a 1. This is because when we go off the end of the right end of the panel, we circle back to its left end. To the left of the 0 we also have a 1. Since these two match, we follow Rule A, flipping all three switches

001

Next, we apply the switching rule to the 0 in the middle of the panel. To the left of this is a 0, thus we match Rule D. Thus, we flip the left (of the current) switch to obtain:

101

Finally, we apply the switching rule to the 1 at the left of the panel. To the left of this is a 1, since when we look to its left, we wrap around to the right end of the panel. This means we must apply Rule D again and flip the left (of the current) switch to obtain:

100