Enemy at the Gates

You are building a circuit which consists of a hierarchy of logic gates which take a series of 2^n bits as input. The bits are processed in stages. The first stage pairs all the bits from the input and applies a logic operation to each pair, leaving 2^{n-1} bits for the next level. The process is repeated at each subsequent level until only 1 bit remains. You have decided to write a program to aid you in the design of this circuit that will determine for a given set of input bits and a given logic gate layout, what the output bit will be.

Below is a description of each type of logic gate that your circuit can consist of:

AND GATE - Outputs a 1 if both bits are 1, outputs 0 otherwise.

OR GATE - Outputs a 1 if at least one of the bits is 1, outputs 0 otherwise.

XOR GATE - Outputs 0 if both bits are equal, outputs 1 otherwise.

Below is a schematic of a possible circuit that could be input into your program. It corresponds to the second data set in the sample input:

```
1 -
  A - 0 -
0 -      |
        X|- 0 -
0 -      |     |
  A - 0 -      |
1 -            |
              X|  - 0
1 -            |
  R - 1 -      |
1 -      |     |
        A|- 0 -
0 -      |
  R - 0 -
0 -
```

Input

The first line of input will contain an integer number indicating how many data sets are to be processed. The remaining lines will contain the data sets. Each data set will consist of three lines formatted according to the following descriptions:

Bit Count - An integer corresponding to the number of bits being fed into the circuit. This integer will be a number between 2 and 32 and will be a power of 2.

Bit List - A space separated list of bits (either 0 or 1) that will be fed into the circuit.

Gate List - A space separated list of logic gates. The first $n/2^1$ gates constitute the first stage and operate directly on the input bits, the next $n/2^2$ gates constitute the second stage and operate on the results from the first stage, and so on for the other stages until only one gate remains. (Note, it is possible the there is only one stage) Each character in the list will either be 'A', 'R', or 'X' (corresponding to AND, OR, or XOR gates respectively). There will be exactly *n*-1 logic gates (where *n* is the number of bits read in).

Output

The output will consist of the bit (0 or 1) resulting from the logic gate operations on the input bits. Each pair of bits will be fed into the first $n/2^1$ gates, and the results of each of those operations will be fed into the next $n/2^2$ gates, and so on until the result is reduced to a single bit (0 or 1).

Example

Input:

```
2
2
1 1
A
8
1 0 0 1 1 1 0 0
A A R R X A X
```

Output:

```
1
0
```