
Digital Circuits

P26615_en

Novè Concurs de Programació de la UPC - Final (2011-09-21)

Taking into account the large number of telecommunication students that have participated in “El Concurs de Programació”, Professor Oak started getting interested in digital circuits. Digital circuits are composed of gates AND, OR and XOR, each with two input pins and one output pin. Every pin has several bits, so we can consider them to be just base-2 numbers. In terms of C++ code, these gates do the following bitwise operations:

AND: `out = in1&in2;` OR: `out = in1|in2;` XOR: `out = in1^in2;`

To simplify things, we focus on circuits with a tree structure: The two inputs of a gate are connected to exactly two outputs or none. Moreover, every output but one is connected to exactly one input. The single output not connected to any gate will be called the “external output”. All inputs not connected to any gate will be called the “external inputs”.

Evaluating the output of a circuit like this is straightforward, but with noise things get more complicated. To model this noise, we assign to each external input an interval $[\ell, r]$, meaning that it can receive any value between ℓ and r with uniform probability.

Given the description of a circuit, please compute the expected value of its external output.

Input

Input consists of several cases. Each case begins with the number of gates n . Follow, in order, the information of the gates. For every gate i between 0 and $n - 1$, first we have “AND”, “OR” or “XOR”. Then, we have an ‘E’ when both inputs are external, or a ‘C’ when both inputs are connected to outputs. After an ‘E’ follow the range of possible values for the first input $[\ell_1, r_1]$, and the range for the second input $[\ell_2, r_2]$. After a ‘C’ follow the two indices of the gates connected to i . Assume that n is between 1 and 1000, and that the external inputs are all in the range $[0, 10^5]$.

Output

For every case, print the expected value of the external output of the circuit with four digits after the decimal point. The input cases have no precision issues.

Sample input

```
1
AND E 1 1 1 1
1
AND E 0 1 0 1
1
OR E 0 0 0 0
1
OR E 0 1 0 1
1
XOR E 0 1023 0 1023
3
AND E 1 1 0 0
OR E 0 0 1 1
```

```
XOR C 0 1
3
XOR C 2 1
AND E 112 300 20 50
OR E 30 45 67 80
```

Sample output

1.0000
0.2500
0.0000

```
| 0.7500  
| 511.5000  
| 1.0000  
| 99.7958
```

Problem information

Author : Josep Angel Herrero
Generation : 2024-04-30 18:03:38

© *Jutge.org*, 2006–2024.
<https://jutge.org>