PROBLEM: Given string data input representing a mathematical expression, determine the possible location(s) of the missing symbol of enclosure. Although we show spaces between characters for readability, the input strings will have no spaces. All operands will be integers less than 1000. The order of evaluation for the symbols is parentheses, brackets and then braces.

Given **{ [(2 + 3) * 6 / 10 }.** A bracket is missing. It could be correctly placed in several locations:

{ [(2 + 3)] * 6 / 10 }   **Location 8**

{ [(2 + 3) * 6] / 10 }   **Location 10**

{ [(2 + 3) * 6 / 10] }   **Location 13**

INPUT: There will be five lines of input. Each line will contain a string of characters with no spaces representing a mathematical expression. Each type of symbol of enclosure will be used at most once. Each expression will have one missing symbol of enclosure. The operators used will be: +, −, * and /.

OUTPUT: For each line of input list all the locations in that expression where the missing symbol of enclosure can be correctly placed. Note: single integers are never enclosed.

SAMPLE INPUT	SAMPLE OUTPUT
1. { [(2 + 3) * 6 / 10 }	1. 8, 10, 13
2. { [7 * 25 / 5 + 14) * 51] + 6 }	2. 3, 5, 8
3. { 60 + [15 * (520 − 505)] / 5 − 23	3. 19, 21, 24
4. { 32 * 510 + (8 / 4) * 2 + 43] / 24 }	4. 2, 5, 9, 15
5. { [(24 + 900 / 300 * 64] }	5. 10, 14, 17

TEST DATA

TEST INPUT	TEST OUTPUT
1. 2 * (3 + 6 * 10 / 2 / 36	1. 7, 10, 12, 15
2. [16 * (4 − 3) − 4 / 2 + 10	2. 10, 12, 14, 17
3. 2 + 4 * 5 / 10) + 6 * 3	3. 1, 3, 5
4. 200 / [(2 * 4) / 8] } * 5	4. 1, 5
5. 10 − { 4 * 5 + 20] / (2 + 6) * 12 − 8 }	5. 5, 7