acm International Collegiate Programming Contest 2013 IBM event sponsor

ACM International Collegiate Programming Contest

2013

Latin American Regional Contests

November 8th-9th, 2013

Warmup Session

This problem set contains 2 problems; pages are numbered from 1 to 2.

This problem set is used in simultaneous contests hosted in the following countries:

- Argentina
- Bolivia
- Brasil
- Chile
- Colombia
- Cuba
- México
- Perú
- República Dominicana
- Venezuela

v1.0

General Information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename*`.c`, *codename*`.cpp` or *codename*`.java`, where *codename* is the capital letter which identifies the problem.

Input

1. The input must be read from standard input.
2. The input consists of a single test case, which is described using a number of lines that depends on the problem. No extra data appear in the input.
3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.
4. Every line, including the last one, has the usual end-of-line mark.

Output

1. The output must be written to standard output.
2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data must appear in the output.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. Every line, including the last one, must have the usual end-of-line mark.
5. To output real numbers, round them to the closest rational with the required number of digits after the decimal point. Test case is such that there are no ties when rounding as specified.

Problem A

Arithmetic with Morse

Morse code is a method used to transmit text messages as a series of dots "`.`" and dashes "`-`". For example, the letter "`A`" is represented with "`.-`" and the letter "`B`" with "`-...`". This code has been used for several years in the army and civil applications, but you are going to use it to do math.

With this in mind, we assign values to dots and dashes, and to save space we use two additional characters. The following table shows the four allowed characters and their values.

Character	Value	
.	1	
-	5	
:	2	(two times ".")
=	10	(two times "-")

A Morse number is a string that only contains the above four characters; its value is the sum of the values assigned to each individual character. As an example, the value of "`=.-..`" is $10+1+5+1+1 = 18$. Notice that each Morse number represents a unique value, but there are values that can be represented with several Morse numbers. For instance, there are three Morse numbers with value 3: "`...`", "`.:`" and "`:.`".

Well, numbers are ready. To form expressions we also need operators. We consider two arithmetic operators: "`+`" represents addition, while "`*`" represents multiplication. A Morse expression is a sequence of strings that interleaves Morse numbers and operators, that starts and ends with a Morse number, and contains at least one operator. Morse expressions can be evaluated by replacing each Morse number by its value, and then evaluating the obtained mathematical expression using the common operators precedence and associativity. For example, the value of the Morse expression "`=.-.. + ... * :.`" is $18 + 3 \times 3 = 18 + (3 \times 3) = 27$. Given a Morse expression, print its value.

Input

The first line contains an integer N ($1 \leq N \leq 4$) representing the number of operators in the Morse expression. The second line contains $2N + 1$ non-empty strings representing the Morse expression. The line interleaves Morse numbers and operators, being the first and last strings Morse numbers. Each Morse number is at most 7 characters long, where each character is either "`.`", "`-`", "`:`" or "`=`". Each operator is either "`+`" or "`*`".

Output

Output a line with an integer representing the value of the Morse expression.

Sample input 1	**Sample output 1**
2 =.-.. + ... * :.	27

Sample input 2	**Sample output 2**
3 - * - + - * -	50

Problem B
Beautiful Yekaterinburg

Yekaterinburg is a beautiful city founded in the eighteenth century. Your task in this problem is to give us information about the precise year of its foundation. To make the task less overwhelming, you are only asked to calculate one of the four digits of the year.

To indicate the position of the digit you must calculate, digits are numbered between 1 and 4, from the most significant to the least significant. As an example, for the year 2013 digit 1 is "2", digit 2 is "0", digit 3 is "1" and digit 4 is "3".

In case you failed to bring an encyclopedia with you, information about Yekaterinburg may be obtained via a clarification.

Input

A single line that contains an integer D $(1 \leq D \leq 4)$ indicating the position of the digit you must calculate.

Output

Output a line with an integer representing the digit at position D of the year Yekaterinburg was founded.

Sample input 1	**Sample output 1**
`1`	`1`

Sample input 2	**Sample output 2**
`2`	`7`