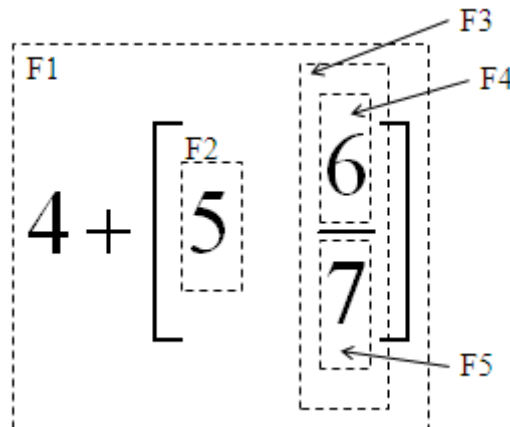


F. Formula Editor

In this problem, you're to write a formula editor. Technically, a formula is an expression, which is a sequence of elements. There are 3 kinds of elements: basic elements (arithmetic operator, parenthesis, digit and letters), matrices and fractions (discussed below).

The editor builds an invisible box for the expression being edited. Since the cells of a matrix are also expressions, each cell has a box enclosing it. Similarly, the denominator and numerator of a fraction are both expressions, so each of them has a box.



In the expression above, there are 5 boxes. F1 encloses the whole expression, F2 and F3 enclose two matrix cells, F4 encloses the numerator and F5 encloses the denominator.

It's not difficult to see that boxes are nested. If box A directly contains box B, we say box A is the parent box of box B (for example, F1 is the parent of F2 and F3, and F3 is the parent of F4 and F5; If box A and box B have the same parent box, we say they're sibling boxes (for example, F4 and F5 are siblings and F2 and F3 are siblings, too).

Cursor Movement

At any time, the cursor is always directly contained in a box. It might be to the left of all the elements in the box (i.e. at head position), to the right of all the elements (i.e. at tail position), or between two consecutive elements. If the cursor is between element x and y, and x is to the left (so y is to the right), we say the cursor's immediate left element is x, and immediate right element is y.

The cursor supports six kinds of movements: HOME, END, and four arrow directions. Let A be the inner-most box that contains the cursor, then

Home(End): place the cursor at the head(tail) position of A (it's still directly in A!).

Up(Down): If A has a sibling box to its up(down) direction, place the cursor at the head position of B, otherwise check A's parent box (if A's parent box has a sibling...). If none of A's ancestor boxes have such a sibling box, do nothing for this command.

Left(Right): There are four cases.

- I If the cursor is at the head(tail) position of A, then place the cursor at the tail(head) position of A's left(right) sibling B. If there is no such B, place the cursor directly in the A's parent box C, to the immediate left(right) of A.
- I If the cursor's immediate left(right) element is a fraction, place the cursor at the tail(head) position of the numerator.

- I If the cursor's immediate left(right) element is a matrix with n rows and m rows, then place the cursor to the tail(head) position of the box at row $\lfloor n/2 \rfloor$ and column 1(column m).
- I If the cursor's immediate left(right) element is a basic element, then place the cursor to the immediate left(right) of that element.

Output Formatting

The formula is output in ASCII format. Each box is formatted as an ASCII rectangle (though most of them are spaces), which is the horizontal concatenation of the ASCII rectangles of all its elements (inner rectangles). Inner rectangles are aligned with their base lines (will be explained shortly). There are no spaces between consecutive elements, and there are no extra lines or columns between the inner rectangles and the boundary of the outer rectangle.

Each element is also formatted as an ASCII rectangle:

- I Basic elements occupy exactly one line, which is also the base line. We use three characters “ - “ (that is, one space at each side) to represent the subtraction operator, and all other elements is formatted as a single character.
- I A matrix is formatted as follows: first, format all the boxes of the matrix cells, and then arrange them into a matrix. The boxes in the same row are aligned according to the base lines, and the boxes in the same column are aligned horizontally. Consecutive rows are separated by an empty line, and consecutive columns are separated by an empty column. Finally, a pair of square brackets is added to the both sides of base lines for each row. The base line of the whole matrix is the base line of the center row if there are an odd number of rows. Otherwise, the base line of the whole matrix is the center empty line.
- I A fraction is formatted as follows: first, format the denominator's box and the numerator's box, then draw a horizontal line (a sequence of ‘-‘ characters) between these two, which is also the base line of the whole fraction. The width of the line is the larger value of two widths, plus 2 (i.e. one more ‘-‘ on both sides). The denominator and the numerator are aligned horizontally.

Note that when aligning rectangles horizontally, we fix the position of the widest rectangle, and try to move other rectangles so that their center columns have the same horizontal position of the widest rectangle. When this is not possible (i.e. for rectangles whose width has a different parity from the widest one), we can move these troublesome rectangles 0.5 unit to the left of the ideal position, like this:

```

      XXXX
      -----
      XXX

```

There is a special case: if the expression is empty, the formatted ASCII rectangle is an empty line, i.e. its width is zero, but its height is one. Naturally, the empty line is the base line.

Input Handling

The input of the editor is already converted to a sequence of command strings. For each string:

- I If it contains a single character, it must be a basic element. Insert that element at the cursor, and move the cursor to its immediate right.
- I If it is `Matrix` or `Fraction`, then insert an 1×1 matrix or an empty fraction at the cursor, then move the cursor right once. Note that, before moving the cursor right, the new matrix/fraction is to immediate right of the cursor.
- I If it is `AddRow` or `AddCol`, insert a row/column before the box that *directly* contains the cursor and place the cursor in the new row/column of the same column/row. If the box is not a matrix cell, check its parent box, until a matrix cell is found. If there is no matrix cell containing the cursor, ignore the command.
- I If it is one of `Home`, `End`, `Left`, `Right`, `Up`, `Down`, follow the cursor movement rules above.

Input

There will be several test cases. Each test case contains a series of command strings, one in each line, ending with command **Done** followed by an empty line. Note that the resulting formula is not necessarily mathematically correct. There will be at most 1000 commands in each test case, and the whole input size does not exceed 200KB.

Output

For each test case, print the final formula in ASCII form. Print the case number in the first line, then an empty line after the formula. Don't print any trailing spaces, but don't omit empty lines (e.g. there is an empty line at the end of sample output 3).

Sample Input

```
-
5
Fraction
1
Down
6
Right
*
Matrix
AddCol
AddCol
1
Right
2
Right
3
Right
*
Matrix
AddRow
AddRow
1
Down
2
Down
3
Done

1
+
Fraction
1
Down
1
+
Fraction
1
Down
1
+
Fraction
1
```

Down
x
Up
Up
Right
Right
Home
Up
Done

Fraction
a
Done

Matrix
Fraction
a
Down
b
Matrix
Fraction
c
Down
d
AddRow
e
Done

Output for Sample Input

Case 1:

$$\frac{1}{5} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \frac{1}{6} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Case 2:

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{x}}}}$$

Case 3:

$$a \dots$$

Case 4:

$$\frac{a}{b} \begin{bmatrix} \dots \\ e \end{bmatrix}$$

$$\begin{array}{c} c \\ [---] \\ d \end{array}$$

Hint

This problem is tricky. Please make sure your program can pass the test cases in the gift package in the contest website. Don't forget that clarification requests to my email address are always welcome.