Problem

This problem is to write a program to fill a region in a rectangular array of pixels given the outline of the region.

Input

The input is a sequence of problem instances. Each problem instance begins with a line containing the row-count of the array, the column-count of the array and the number-of-regions to fill as decimal integers. Rows are counted from top to bottom beginning with 1. Columns are counted from left to right beginning with 1. The input ends with a row-count of 0. Row-count will be at most 47 and column-count will be at most 63.

The first line of each problem instance is followed by region descriptions for number-of-regions regions. Each region description begins with a line containing: a single character to be used to fill the region, the row number of the start pixel, the column number of the start pixel and the number of pixels in the boundary which will always be at least two. This line is followed by lines of direction codes (up = A, up right = B, etc.):

H A B  
G     C  
F  E  D

describing the outline traversed clockwise. The start pixel may be any point on the outline.

Output

For each problem instance, the output consists of row-count lines of column-count characters each of which is either a period (.) indicating than no region includes that pixel or the fill character of the region which includes that pixel. The row-count lines are followed by a single blank line. This array may be preceded by one or more lines of the following form (where A or B are the fill characters specified in the region header):

REGION A GOES OUTSIDE THE ARRAY
If the boundary path goes outside the boundaries of the array specified for the problem instance.
REGION A BOUNDARY IS NOT CLOSED
If the boundary as specified does not return to the start point.
REGION B BOUNDARY INTERSECTS REGION A
If the boundary of region B contains points of a previously specified region. If a previous region fails any of these tests it is not considered for intersection with following regions.

For each region, the first condition to be violated is to be displayed. Any region for which an error line is given will not be filled in the rectangular array.
## Example

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 40 4</td>
<td>........................................</td>
</tr>
<tr>
<td>B 3 21 22</td>
<td>........................................</td>
</tr>
<tr>
<td>C 5 8 36</td>
<td>.AA.....................................</td>
</tr>
<tr>
<td>D 10 24 38</td>
<td>.A......................BBB........BBB........</td>
</tr>
<tr>
<td>CCCCCCDEEEGFEDCCEEEGGGGGAAAAACCBAAHGAAA</td>
<td>.....................BBB....BBB........</td>
</tr>
</tbody>
</table>
| A 2 2 3 | ...............CCC............BBB..BBB..........
| CEH | .................................................................. 
| 10 20 4 | .................................................................. 
| A 4 6 10 | .................................................................. 
| GGAAACCEEE | .................................................................. 
| B 6 16 30 | .................................................................. 
| CCCCCCCCCCCCCCGCCCEGgggggggggg | ..................................................................
| C 5 6 10 | .................................................................. 
| CCCCCDDFFFG | .................................................................. 
| D 6 2 10 | .................................................................. 
| AAACCEEggg | .................................................................. 
| 0 0 0 | .................................................................. 

REGION B GOES OUTSIDE THE ARRAY
REGION C BOUNDARY IS NOT CLOSED
REGION D BOUNDARY INTERSECTS REGION A

...AAA..............
...AAA..............
...AAA..............
...AAA..............
....................
....................
....................
....................
....................
....................

...AAA..............
...AAA..............
...AAA..............
...AAA..............
....................
....................
....................
....................
....................

...AAA..............
...AAA..............
...AAA..............
...AAA..............
....................
....................
....................
....................
....................