# Plane Truth

This is an exercise with planar graphs. Simply devising the data structures and algorithms to do it is sufficient. Concern about time efficiency is not. The idea is to build and explore planar graphs. Submit your solution under the file name *PlaneTruth* with the appropriate language extension.

## Input

There may be multiple planar graphs to explore. Each case consists of the description of a planar graph followed by an exploration of it. The last case is followed by a line containing 0 0 0 (three zeroes).

The first line of a case contains the number of faces $f$, then vertices $v$, and then edges $e$. Faces are numbered from 1 through $f$ in the order described. Each of the next $f$ lines contains the description of a face. A face is described by a string of uppercase letters representing the sequence of vertices encompassing the face in clockwise order, starting with an arbitrary vertex. The perimeter of the graph is considered to bound the infinite exterior which is also considered to be a face. It is the last face to be described, also noting that clockwise is from the perspective of a person inside a face which in this case is outside the graph.

Edges are designated by pairs of uppercase letters representing the vertices each connects. Each edge can be thought of as a pair of directed edges pointing in opposite directions and connecting the same vertices. The right hand side of each such directed edge bounds the interior face and each left hand side bounds some exterior face.

Following the description of a planar graph will be some paths represented by strings of vertices. Each path is specified on a single line. The last path of a case is followed by the first line of the next case (or the end of input) which of course consists of three integers. Paths are of arbitrary length and vertices may be repeated. There may be reversals in a path.

## Output

For each new planar graph, print its number formatted as shown in the example. For each path of exploration of a given planar graph, print its number formatted as shown in the example. At each vertex along the path, print the name of the vertex followed by a sorted list of the adjacent vertices. At each edge along the path, print the number of the face on the right followed by the number of the face on the left. All should be formatted as in the example.

## Sample Input

```
4 6 8
ACEB
DFCA
ECF
FDABE
FCEB
DAD
0 0 0
```

## Sample Output

```
graph 1
path 1
F CDE
  2 3
C AEF
  1 3
E BCF
  1 4
B AE
path 2
D AF
  4 2
A BCD
  2 4
D AF
```