Diamonds in the Rough

Filename: `diamond`

Far off in Ascii Land a new high tech diamond mining company is looking to obsolete their competition. In Ascii Land diamonds are made out of forward slashes and backslashes, but they must be found in the correct symmetrical configuration. Informally, the slashes must make a 'V'-shape with an equal size '∧'-shape on top of it. In the past, after a slab of text was mined and cleaned of all unnecessary character types, the extraneous slashes needed to be removed by hand. This new mining company plans to automate the process.

Diamonds contain no less than two forward slashes and two backslashes and always contain an equal number of forward slashes and backslashes. For example:

```
../\............../\......          ../\..../\......./\......
..\/.../\....../..\.....          .../.../..\..../..\.....
....../..\....././\.\....          ....../....\./....\.....
......\../....\.\/./....          ......\.../..\..../....
.......\/......\../.....          ........\/.....\../.....
................\/......          ................\/......
```

Four valid diamonds   No valid diamonds

Diamonds are capable of intersecting, overlapping, and containing other diamonds (as shown above; other examples are shown in the Sample Input).

The Problem:

Given a slab of text containing characters '.'(period), '\' (backslash), and '/' (forward slash), output the same slab of text with any forward slash or backslash that is *not* a part of any valid diamond replaced with a period.

The Input:

The first line of the input contains a single, positive integer, s, representing the number of slabs in the input. This is followed by s descriptions of slabs. Each description starts with a line containing two integers, h and w ($1 \le h \le 50$; $1 \le w \le 50$), representing the height and width of the slab, respectively. On each the following h lines, there will be w characters, depicting the slab. All slab characters are only '.' (period), '\' (backslash), or '/' (forward slash).

The Output:

For each slab, output a line of the form "`Slab #i:`" where i represents the number of the group you are processing (starting from 1) followed by h lines of width w containing the depiction of the slab with all extraneous slashes removed (replaced by a '.').

Sample Input:

```
3
2 8
\/ /\\/ /\
\/ \/ /\/ \
6 8
/ . / /\\ . \
. / / /\\\ .
/ / / /\\\\
\\/ /\\/ /
. \\/ \/ / .
\ . \\/ / . /
10 24
. . / / / . . / . . \/ \\\\ . . . . / . . .
. . . . . \/ . /\ . \ . . . /\ . . / . . . .
. . \ . / . \/ . /\/ . . /\ . \/ . . . . .
. / . \ . . . \/ \/\ . /\\\/ . . . . . /
. . /\/ \ . . \/ . . / . /\ . \ . . . . / .
. / . /\ . \ . / . . / /\\/ . /\ . . / . .
. \ . \/ . /\ . . . \\/ /\ . \/ . / . . .
. . \/ \/ . . \ . . . \ . \/ . /\/ . . . .
. . / . / . . . . . . . . \ . . / . / . . . . .
. / . / . . . . . . \ . . . \/ \/ . . . . . .
```

Sample Output:

```
Slab #1:
. . /\ . . . .
. . \/ . . . .
Slab #2:
. . . /\ . . .
. . / . . \ . .
. / . /\ . \ .
. \/ . . \/ .
. . \/ \/ . .
. . . \/ . . .
Slab #3:
. . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . /\ . . . . . . . . . . . . .
. . . . . . . / . /\ . . . /\ . . . . . . . .
. . . . . . . \/ \/ . . / . . \ . . . . . . .
. . /\/ \ . . \/ . . / . /\ . \ . . . . . .
. / . /\ . \ . . . . / /\\/ . /\ . . . . .
. \ . \/ . / . . . . \\/ /\ . \/ . . . . .
. . \/ \/ . . . . . . \ . \/ . / . . . . . .
. . . . . . . . . . . . . . \ . . / . . . . . . .
. . . . . . . . . . . . . . . \/ . . . . . . . .
```

18