It's the Great Pumpkin Patch Problem
input file: `pumpkin.in`   output file: `pumpkin.out`

It's almost Halloween and Linus is setting out to the garden to wait for the Great Pumpkin. Unfortunately, due to diversification, there are lots of other gourds in the garden this year, so he needs you to write a program to tell him how many patches of pumpkins there are and how big they are.

For example, consider the following 10 by 10 garden, with zucchini (z), yellow squash (y), spaghetti squash (s), and pumpkins (p):

p	z	z	z	z	z	z	z	z	p
p	y	y	p	z	z	z	z	z	y
p	p	p	p	s	s	s	s	y	y
s	s	s	p	s	s	s	s	y	y
s	s	s	s	p	p	s	s	y	y
s	s	s	s	p	p	s	s	p	y
z	z	z	z	z	z	s	s	p	y
z	z	z	z	z	z	s	s	p	y
y	y	y	y	p	z	s	s	p	y
y	y	y	y	p	p	p	p	p	y

This garden has four patches of pumpkins: one at the top left corner covering 8 squares, one in the top right corner covering 1 square, one in the center covering 4 squares and one near the bottom right covering 10 squares. These patches are shown shaded. Note that in order for a square to be a part of a patch, it must connect with another square in that patch along an edge, not just at a corner (so the squares in the center patch are not part of the patch at the top left corner).

Input

The input to this program will be a number of different gardens. The first line of the input for each garden will be the dimensions of the garden, r, the number of rows in the garden, and c, the number of columns, where $0 \le r \le 40$ and $0 \le c \le 40$. Following the dimensions will be r lines with c characters on each line. Each of these characters will be a lower case letter representing the type of gourd grown in the square. A lower case 'p' will represent pumpkins.

A garden with 0 for the number of rows and/or columns indicates the end of input and should not be processed.
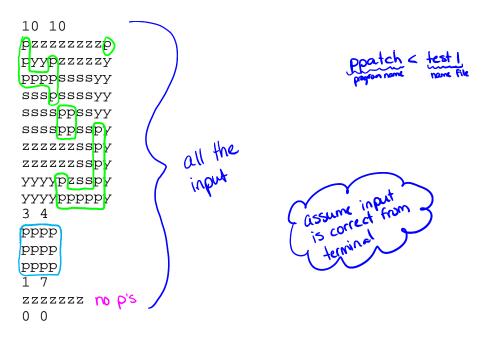
Output

For each garden, output the number of the garden (with the first input set being garden 1), the number of pumpkin patches in the garden, and the size of the pumpkin patches in order from smallest to largest. If there is more than one patch of a given size, print the size as many times as it occurs. Use the following format:

```
Garden # 1: 4 patches, sizes: 1 4 8 10
```

Have a blank line between each line of output.

Sample input

The first input set represents the garden in the garden description.

```
10 10
pzzzzzzzzp
pyypzzzzzy
ppppssssyy
ssspsssyy
sssppssyy
ssspppssyy
ssspssy
zzzzzzssp
zzzzzzssp
yyyypzssp
yyyyppppp
3 4
pppp
pppp
pppp
1 7
zzzzzzz
0 0
```

[handwritten annotations: "all the input" with bracket; "no p's" pointing to zzzzzzz; "ppatch < test 1 — program name, name file"; cloud: "assume input is correct from terminal"]

Sample output (corresponding to sample input)

```
Garden # 1: 4 patches, sizes: 1 4 8 10

Garden # 2: 1 patches, sizes: 12

Garden # 3: 0 patches, sizes:
```

[handwritten: "sorted" underlining "1 4 8 10"]