A prominent microprocessor company has enlisted your help to lay out some interchangeable components (widgets) on some of their computer chips. Each chip's design is an $N \times N$ square of slots. One slot can hold a single component, and you are to try to fit in as many widgets as possible.

Modern processor designs are complex, of course. You unfortunately have several restrictions:

- Some of the slots are disabled.
- Some of the slots are already occupied by other components and cannot be used for widgets.
- There are sibling memory buses connected to the horizontal and vertical edges of the chip and their bandwidth loads need to match. As such, there must be exactly as many components in the first row as in the first column, exactly as many in the second row as in the second column, and so on. Component counts include both the components already specified on the chip and the added widgets.
- Similarly, the power supply is connected at the end of each row and column. To avoid hot spots, any given row or column must have no more than A/B of the total components on the chip for a given A and B.

A specification for a chip is N lines of N characters, where '`.`' indicates an open slot, '`/`' indicates a disabled slot, and '`C`' indicates a slot already occupied by a component. For example:

```
CC/..
././/
..C.C
/.C..
/./C/
```

If no more than 3/10 of the components may be in any one row or column, the maximum number of widgets that can be added to this 5×5 chip is 7. A possible arrangement is below, where '`W`' indicates a widget added in an open slot.

```
CC/W.
W/W//
W.C.C
/.CWW
/W/C/erb
```

Input

The input consists of several test cases. Each case starts with a line containing three integers: The size of the chip N ($1 \leq N \leq 40$), and A and B ($1 \leq B \leq 1000, 0 \leq A \leq B$) as described above. Each of the following N lines contains N characters describing the slots, one of '`.`', '`/`' or '`C`', as described above.

The last test case is followed by a line containing three zeros.

Output

For each test case, display a single line beginning with the case number. If there is a solution, display the maximum number of widgets that can be added to the chip. Display '`impossible`' if there is no solution.

Follow the format of the sample output.

Sample Input

```
2 1 1
/.
//
2 50 100
/.
C/
2 100 100
./
C.
5 3 10
CC/..
././/
..C.C
/.C..
/./C/
5 2 10
CC/..
././/
..C.C
/.C..
/./C/
0 0 0
```

Sample Output

```
Case 1: 0
Case 2: 1
Case 3: impossible
Case 4: 7
Case 5: impossible
```