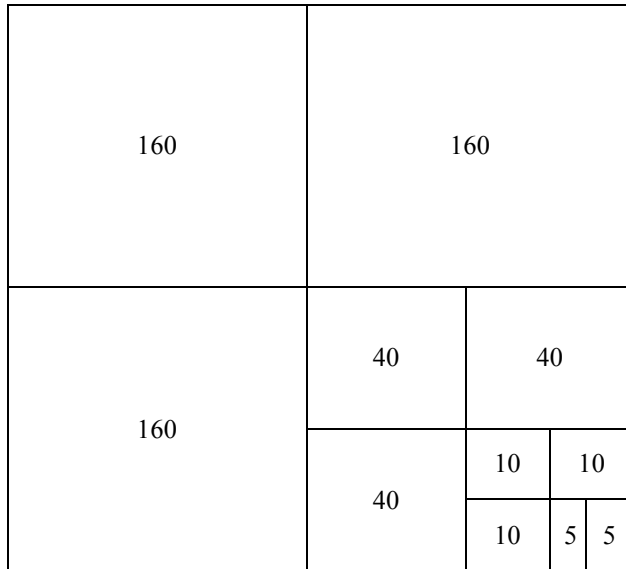

10. Prospectors

Program Name: Prospectors.java

Input File: prospectors.dat

Rob is a surveyor who is plotting the sections of land in an unmapped area of Alaska. A section of land covers one square mile and contains 640 acres of land. Rob has divided the section he is surveying into 128 horizontally or vertically contiguous, congruent, rectangular plots of land each containing 5 acres. Below is a diagram of a 640 acre section of land that has been subdivided as noted in the diagram and chart below.



Name	Acres	Shape
Section	640	Square
Quarter section	160	Square
Quarter quarter section	40	Square
$\frac{1}{4}$ quarter quarter section	10	Square
Plot of land	5	Rectangular

To visualize his survey on paper, Rob has created a section grid that is 8 rows by 16 columns. Each cell of the grid represents one 5 acre plot of land. Some prospectors from the big Alaska gold rush have already staked claims on some of the 5 acre plots of land. Rob decided to denote those plots with capital letters of the alphabet using a unique letter of the alphabet for each prospector. You are to write a program that will tell how many acres are in each of the lots that lay unclaimed. A lot is a piece of land that contains all of the unclaimed plots that are completely surrounded by either the border of the section or by a prospector's claim.

Input

The first line of input will contain a single integer n that indicates the number of surveys to follow. Each survey will contain one section grid of 8 rows with 16 characters in each row. The characters in each row will be a capital letter (A-Z) indicating who owns the plot of land, or an asterisk (*) indicating an unclaimed plot of land. There will be a blank line following each section grid except for the last section grid.

Output

For each survey, you will output on a single line the number of acres available in each unclaimed lot from largest to smallest and separated by a space.

Example Input File (cont. on next page.)

10. Prospectors (cont.)

Example Input File

2

```
**AAAA**AA**B
*****AAAAAA**B
CCCCCDDD*****
*****HHHHHHHH
*****FFFFF**H*
LL**FFFFFF*****
**WW*****R**
**WW*****ZZZZ
```

```
*****
*****YXXXXXX**
*****BB*****
HHHHHHHHHHHHHHH
**HHH*****D*****
****DDDD**DD****
****DD****DDD**
****DD****DDDD**
```

Example Output to Screen

```
135 110 65 35 20
195 75 70 70
```