

Toga Party

Input file: party.in

The guys at Delta House are at it again. They've decided to throw the ultimate toga party. Since they want this to be the best party Faber College has ever seen, they've decided to match up the men and women attending the party in such a way as to maximize the "happiness quotient". The happiness quotient contributed by a pairing is determined by the following sequence of rules:

1. If a pairing has a man and a woman who like each other, a happiness quotient of 4 is contributed to the party.
2. Otherwise, if a pairing has a man and a woman who at least tolerate each other, a happiness quotient contributed is 3.
3. Otherwise, if the woman in a pairing at least tolerates the man, the happiness quotient contributed is 2.
4. Otherwise, if the man in a pairing at least tolerates the woman, the happiness quotient contributed is 1.
5. In all other cases, no happiness quotient is contributed.

So, if Fred, Bob, and Greg were at the party with Janet, Mary, Cindy, and Dianne, and the following preference facts were known:

- Fred likes Janet
- Fred likes Mary
- Fred tolerates Cindy
- Bob tolerates Janet
- Greg tolerates Mary
- Janet likes Fred
- Mary tolerates Fred
- Janet tolerates Bob
- Mary likes Bob
- Mary tolerates Greg
- Cindy tolerates Fred
- Cindy likes Bob

the highest happiness quotient that the party could have would be 9, obtained by matching Fred and Janet (4 points) and Greg and Mary (3 points), and Bob and Cindy (2 points).

Given a number of preference facts, you should determine the maximum happiness quotient that is possible for a party. Each person can be matched with at most one other person at the party. There may be a different number of men and women.

Input:

The first line of each data set contains a non-negative integer n , which is the number of rules for that party. There will then be n lines, each with three letters on the line, $c1Fc2$. Within a rule, the men are identified by lowercase letters ('a'-'z') and the women are identified by uppercase letters ('A'-'Z'). Character $c1$ and $c2$ will identify a man and woman (in either order). The F is either an 'L' or 'T', with 'L' indicating $c1$ likes $c2$, and 'T' indicating $c1$ tolerates $c2$. (For example, aLB represents "a likes B" and GTg represents "G tolerates g".) Each combination $c1$ and $c2$ will appear at most one time in the input, so you do not need to be concerned with conflicting input (e.g., you will not have both aLB and aTB in the same set, but you may have aLB and BTa, since in the first $c1 = a$ and in the second $c1 = B$). There will always fewer than 9 men, and fewer than 9 women.

The last data set is flagged by the number of rules being zero. This set should not be processed.

Output:

Each data set should generate a single line of output in the form

Party n has a maximum happiness quotient of h

where n is the number of the data set (starting at 1), and h is the maximum happiness quotient.

Sample Input:

```
12
fLJ
fLM
fTC
bTJ
gTM
JLf
MTf
JTb
MLb
MTg
CTf
CLb
4
ATa
ATb
aTA
bTA
6
ALa
CLa
CLb
aTA
aLC
bTC
0
```

Output Corresponding to Sample Input:

```
Party 1 has a maximum happiness quotient of 9
Party 2 has a maximum happiness quotient of 3
Party 3 has a maximum happiness quotient of 6
```

UCF Local Contest — September 2, 2006

Crystal Etching (filename: etch)

When one usually thinks of crystals, they think of chandeliers and other ornamental items. Yet, there are crystals that are components in our every day appliances, such as microwaves. These crystals have to be produced at very specific thicknesses to obtain the desired frequency of oscillation. Part of this process involves creating a generic crystal that is too thick, and then calculating exactly how long that crystal needs to be placed in a chemical etch bath to reduce it to the proper thickness. In particular, if the current frequency of oscillation of a crystal is f_1 , and its target frequency is f_2 , then t , the number of seconds necessary to etch the crystal to achieve the desired frequency satisfies the following equation:

$$\frac{f_2 - f_1}{f_1 f_2} = at + b(1 - e^{-ct})$$

where a , b , and c are all dependent upon the actual chemical composition of the etch bath. (Note: The units of frequency are 1/sec.)

The Problem:

Given the current frequency of a crystal (f_1), its target frequency (f_2), and the values of a , b , and c , for a particular etch bath, calculate the amount of time (t , in seconds), rounded to the nearest hundredth that is necessary to etch the crystal to obtain its desired frequency.

The Input:

The first line of the input file will contain a single positive integer n , the number of test cases in the file. Each of the next n input lines will contain information for a single test case. On each line, the following values, all positive real values, will be separated by spaces: f_1 , f_2 , a , b , and c . It is guaranteed that $100000 \geq f_2 > f_1 \geq 10$.

The Output:

For each input case, print a heading followed by the number of seconds (rounded to the nearest hundredth) necessary to etch the corresponding crystal to achieve its target frequency. Assume that no etch time will exceed 1000000.00 seconds. Leave a blank line after the output for each data set. Follow the format illustrated in Sample Output.

(Sample Input/Output on the next page)

Sample Input:

```
2
500 1000 .001 .001 1
1000 2000 .00025 .0005 .6931472
```

Sample Output:

Crystal #1: 0.57

Crystal #2: 1.00



Problem J

Zones

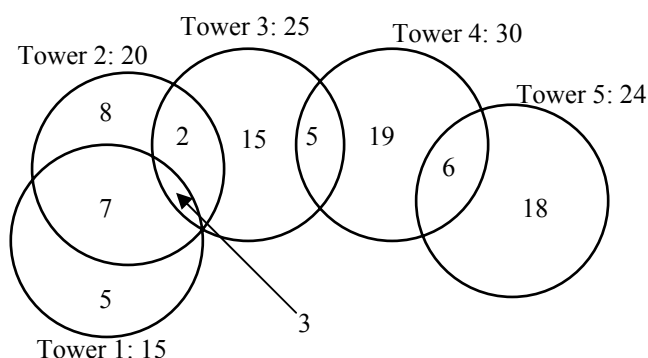
Input File: zones.in

Telephone poles are part of an outdated technology. Cell phones nowadays allow us to call whoever we want, wherever we want, independent of any wire. There is one problem — without a service tower nearby a cell phone is useless.

In the absence of hills and mountains, a service tower will provide coverage for a circular area. Instead of planning where to place the wires, a wireless telephone company has to plan where to build its service towers. Building the towers too far apart causes holes in the coverage and increases complaints. Building the towers too close to each other results in large areas covered by more than one service tower, which is redundant and inefficient.

International Cell Phone Company is developing a network strategy to determine the optimal placement of service towers. Since most customers have replaced their old wired home phones by cell phones, the strategy for planning service towers is therefore to cover as many homes of customers as possible.

The figure below shows the service areas for the five towers ICPC's strategic department planned to build this year. Tower 5 will serve 24 customers, 6 of which are also served by tower 4. Towers 1, 2 and 3 have a common service area containing 3 customers.



Shortly after the plans for these five towers had been published, higher management issued a stop on all tower building. Protesting customers forced them to weaken this decree, and allow the building of three towers out of the five planned. These three towers should serve as many customers as possible, of course. Finding the best possible choice for the towers to build is not trivial (the best choice in this case is towers 2, 4 and 5, serving 68 customers).

You must write a program to help ICPC choose which towers to build in cases like these. If several choices of towers serve the same number of customers, choices including tower 1 are preferable. If this rule still leaves room for more than one solution, solutions including tower 2 are preferable, and so on.

Input

The input file contains several test cases. The first line of each test case contains two positive integers: the number n ($n \leq 20$) of towers planned, and the number of towers to be actually built. The next line contains n numbers, each giving the number of customers served by a planned tower. (The first number refers to the first tower, and so on.) No tower serves more than a million people. The next line contains the number m ($m \leq 10$) of common service areas. Each of the next m lines contains the description of a common service area. Such a line starts with the number t ($t > 1$) of towers for which this is a common service area, followed by the t numbers of the towers. The last number on the line is the number of customers in the common service area. The last line of the input file consists of the two integers 0 0.



Output

For each test case, display the number of customers served and the best choice for the location of the towers. Follow the format of the sample output.

Sample Input	Output for the Sample Input
5 3 15 20 25 30 24 5 2 1 2 7 3 1 2 3 3 2 2 3 2 2 3 4 5 2 4 5 6 5 3 25 25 25 25 25 4 2 1 2 5 2 2 3 5 2 3 4 5 2 4 5 5 5 3 25 25 25 25 25 0 0 0	Case Number 1 Number of Customers: 68 Locations recommended: 2 4 5 Case Number 2 Number of Customers: 75 Locations recommended: 1 3 5 Case Number 3 Number of Customers: 75 Locations recommended: 1 2 3



D: Knitting

Marcia loves to knit. As she knits, she wonders how many stitches the project she is working on will take to complete.

On every project, she starts with a row of a given number of stitches, and then adds more rows. Sometimes the next row will have the same number of stitches as the previous row and other times the next row will have more or less stitches than the previous row.

For example, a pattern for a triangular shawl may begin with just 3 stitches and add 2 stitches on each row. So, the first row will have 3 stitches, the second row will have 5 stitches, the third will have 7 stitches, and so on. If the project has a total of 3 rows, then it has a total of 15 stitches.

A more complex scarf project may have a 4 row repeating pattern that increases 6 stitches on the first row of the pattern, decreases 2 stitches on each of the next two rows, and has no change on the final row of the pattern. So, a scarf that has 50 stitches on the first row will have 56 on the second row, 54 on the third row, 52 on the fourth row, and 52 on the fifth row. On the sixth row, the pattern repeats, so there will be an increase of 6 stitches for a total of 58 stitches on that row. If the project stops there at 6 rows, then it will have a total of 322 stitches.

You will write a program to help Marcia figure out how many stitches a project will take to complete.

Input

The input to your program will be information about one or more projects. Each project's description will take up 2 lines. The first line contains three integers:

N M K

Where **N** ($1 \leq N \leq 100$) represents the number of stitches in the first row of the project, **M** ($1 \leq M \leq 1000$) represents the total number of rows in the project, and **K** ($1 \leq K \leq 100$) represents the number of rows in the repeating pattern. On the following line will be exactly **K** integers, each one in the range from -100 to 100 (inclusive), indicating the repeating pattern, where negative values indicate a number of stitches to decrease, positive values indicate a number of stitches to increase, and 0 indicates no change. In any project, the pattern will never cause any row to have 0 or fewer stitches. End of input is indicated by a line with three 0's.

Output

For each project, give the total number of stitches in the completed project. Print each integer answer on its own line, with no blank lines between answers.



Sample Input

```
3 3 1
2
50 6 4
6 -2 -2 0
0 0 0
```

Sample Output

```
15
322
```



E: Minesweeper

Minesweeper is a game played on a $R \times C$ rectangular board. Some of the cells contain mines, and others are empty. For each empty cell, calculate the number of its adjacent cells that contain mines. Two cells are adjacent if they share a common edge or point. This means that each cell has a maximum of 8 neighbors (up, down, left, right, four diagonals).

Input

There will be multiple test cases. The first line of each test case will have two integers, R and C ($1 \leq R, C \leq 100$), indicating the number of rows and columns of the board. The next R lines each contain exactly C characters. Each character is either a '*' (asterisk) indicating a mine, or a '.' (period) indicating an empty cell. The last data set is followed by a line containing two 0's.

Output

Print each board on R lines with C characters per line, and replace every '.' with the appropriate digit indicating the number of adjacent cells that contain mines. Leave the '*' cells intact. Do not print any whitespace between cells. Do not print any blank lines between answers.

Sample Input

```
3 2
..
.*
..
5 5
*.*.*
..*..
*****
.....
..**.
```

```
0 0
```

Sample output

```
11
1*
11
*3*3*
36*63
*****
24553
01**1
```