2017 Consortium for Computing Sciences in Colleges
Programming Contest
Saturday, November 4th
Furman University
Greenville, SC

There are eight (8) problems in this packet. Each team member should have a copy of the problems. These problems are NOT necessarily sorted by difficulty. You may solve them in any order.

Remember input/output for the contest will be from `stdin` to `stdout`. `Stderr` will be ignored. Do not refer to or use external files in your source code. Extra white space at the end of lines is ignored, but extra white space at the beginning or within text on a line is not ignored. An extra blank line of output is ignored, but blank lines at the beginning or between lines of text are not ignored. Have a lot of fun and good luck! ☺

Problem 1. Solar Eclipse

Problem 2. Successor Matrix

Problem 3. Number Convert

Problem 4. Between Parabolas

Problem 5. Triangle Island

Problem 6. What’s a Hanin?

Problem 7. Handshakes

Problem 8. Overlap Length

Problem 1

Solar Eclipse

The solar eclipse of August 21st, 2017, dubbed "The Great American Eclipse" by the media, was a total eclipse visible within a band across the entire contiguous United States, passing from the Pacific to the Atlantic coasts. Right here at Furman University, a total solar eclipse occurred at exactly 2:38 pm EDT over Paladin Stadium. Prior to this event, no solar eclipse had been visible across the entire contiguous United States since 1918; not since the February 1979 eclipse had a total eclipse been visible from anywhere in the mainland United States. The next total solar eclipse will be seven years from now in April 2024.

Your job is to take the results from a recent survey of random residents in all 50 states on whether or not they witnessed a full eclipse, partial eclipse, or no eclipse at all. Your program should report back how many different states were surveyed, the overall totals reported on the three different possible responses, and the state with the most responses. In the event there is a tie for the state with the most responses, print the one that would come first alphabetically.

Input

The input will consist of one or more strings, one per line. Each string will consist of a state postal code, a blank space, and a single character (`F`, `P`, or `N`) representing full, partial, or no eclipse. The last line of input will be a line with string `-1`, and should not be processed.

Output

Your output should consist of exactly five lines in the format below. All total values will be a nonnegative integer representing the totals computed from the input.

Sample Input

```
SC F
AL P
SC F
TN P
UT P
VA F
SC F
WA F
WV F
WI P
WY N
TN P
TN F
-1
```

Output Corresponding to Sample Input

```
Different States Surveyed = 9
Total Full Eclipses Reported = 7
Total Partial Eclipses Reported = 5
Total No Eclipses Reported = 1
State With Most Responses = SC
```

Problem 2

Successor Matrix

The *successor* of an integer value is the integer that is one greater than that value. For example, the successor of 15 is 16. A *successor matrix* shows the position of the successor of each element in a given 2D integer matrix. The successor matrix has the same dimensions as the given integer matrix. Each element in the successor matrix is the position (row, column) of the corresponding integer matrix element's successor. The largest element in the integer matrix does not have a successor, so its corresponding position in the successor matrix is `null`. You may assume `null` will only occur once in the output. To illustrate the successor relationship, the values 15 and 16 in the integer matrix below are shaded. In the successor matrix, the shaded element shows that the position of the successor of 15 is `(0,2)`. The largest value in the matrix is 19, so its corresponding element in the successor matrix is labeled as `null`. Your job is to write a program which takes a 3 x 4 matrix of integer values, and produces its successor matrix.

	0	1	2	3
0	12	10	16	18
1	11	9	19	17
2	13	15	14	8

	0	1	2	3
0	(2,0)	(1,0)	(1,3)	(1,2)
1	(0,0)	(0,1)	null	(0,3)
2	(2,2)	(0,2)	(2,1)	(1,1)

Input

The first line of input will contain a single integer *n*, which will indicate the number of test cases. You may assume that 1 ≤ *n* ≤ 100. Your input will be in row major order consisting of three lines of four integer values representing a 2D integer matrix. Each integer value in the matrix will be a positive integer *x*, 1 ≤ *x* ≤ 100. The four integers on each line will have a single blank space between. The minimum and maximum integer values in the matrix are guaranteed to be unique, and all ten integers greater than the minimum and less than the maximum will exist in the matrix. For example, the ten integers 9 through 18 are present in the matrix above with a minimum of 8 and maximum of 19.

Output

The output should be in the exact format seen below consisting of a header followed by a successor matrix number. This is followed by three lines of four positions in the format `(row,column)`. Each of the four positions on a line should have one blank space separating each position. Use `null` to indicate the largest element and one with no successor. A blank line should separate the output of each successor matrix.

Sample Input

```
2
12 10 16 18
11 9 19 17
13 15 14 8
22 20 15 11
12 14 13 21
19 16 17 18
```

Output Corresponding to Sample Input

```
Successor Matrix 1:
(2,0) (1,0) (1,3) (1,2)
(0,0) (0,1) null (0,3)
(2,2) (0,2) (2,1) (1,1)

Successor Matrix 2:
null (1,3) (2,1) (1,0)
(1,2) (0,2) (1,1) (0,0)
(0,1) (2,2) (2,3) (2,0)
```

Problem 3

Number Convert

The purpose of this program is to interpret numbers that are expressed in a special letter format. The input consists of strings, one per line. Each string is a word, but it could be a nonsense word. Each character in the string is a letter, or '?' or '!'. Each symbol has a numerical value, and these values need to be added up to produce the value of the string as follows.

Symbol	Value
a	1
b	2
c	3
d	4
e	5
f	6
g	7
h	8
i	9
j	10
k	20
l	30
m	40
n	50
o	60
p	70
q	80
r	90
s	100
t	200
u	300
v	400
w	500
x	600
y	700
z	800
?	900

Symbol	Value
A	1,000
B	2,000
C	3,000
D	4,000
E	5,000
F	6,000
G	7,000
H	8,000
I	9,000
J	10,000
K	20,000
L	30,000
M	40,000
N	50,000
O	60,000
P	70,000
Q	80,000
R	90,000
S	100,000
T	200,000
U	300,000
V	400,000
W	500,000
X	600,000
Y	700,000
Z	800,000
!	900,000

Input

The input will consist of one or more strings, one per line. Each name will consist of 1 to 80 characters. The last line of input will be a line with string `-1`, and should not be processed.

Output

For each line of input generate an output line in the format below. This output line consists of the input string followed by a blank space, a =, a blank space, and the converted string.

Sample Input

```
What
is
my
hash
code?
-1
```

Output Corresponding to Sample Input

```
What = 500209
is = 109
my = 740
hash = 117
code? = 972
```

Problem 4

Between Parabolas

Mary aced differential calculus last quarter. And now she is taking her next math course, which is integral calculus. In a recent lesson, Mary learned that definite integrals are useful to find the area between two curves drawn in the Cartesian plane. Mary's favorite curve is the parabola. To get ready for her next test, she would like to practice working through problems where she needs to find the area between two intersecting parabolas. You will write a program to help Mary check her answers. An instance of this problem consists of two parabolas A and B, defined by:

Parabola A: $y = ax^2 + bx + c$

Parabola B: $y = dx^2 + ex + f$

You may assume that each of the constants a, b, c, d, e, and f is a real number whose value is between 0.01 and 100.0, inclusive, or between -100.0 and –0.01, inclusive. The graphs of parabolas A and B might not intersect at all. Or they might intersect at a single point. In these cases, there is no enclosed region between A and B. However, it is often the case that A and B will intersect in exactly two points. In this case, a finite region lies between the two parabolas vertically, and between the two points of intersection horizontally. Your program is to find the area of this region.

Input

The first line of input will contain a single integer n, which will indicate the number of test cases. You may assume that $1 \le n \le 100$. Each of the next n lines of input will contain one test case. A test case will consist of the real numbers a, b, c, d, e, and f, as described above. These six numbers will be separated from each other by one or more blank spaces. Furthermore, blank spaces may appear at the beginning and/or end of the input lines.

Output

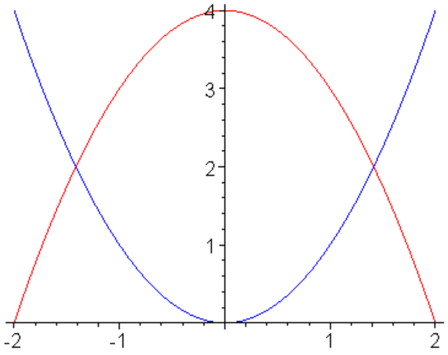
For each test case, your program needs to print the following statement:

- `Segment <number> area is <area between the curves>`

The `<number>` is the number of the test case. Test case numbers begin at 1. The area between the curves must be printed to exactly three decimal places. Your output should look like the example output below. If there is no enclosed region between the parabolas, the area should be reported as 0.000.

Sample Input

```
4
1 0 0 -1 0 8
3 -2 1 1 7 6
1 7 6 3 -2 1
1 0 0 0.3 0 0
```



Output Corresponding to Sample Input

```
Segment 1 area is 21.333
Segment 2 area is 55.458
Segment 3 area is 55.458
Segment 4 area is 0.000
```

Problem 5

Triangle Island

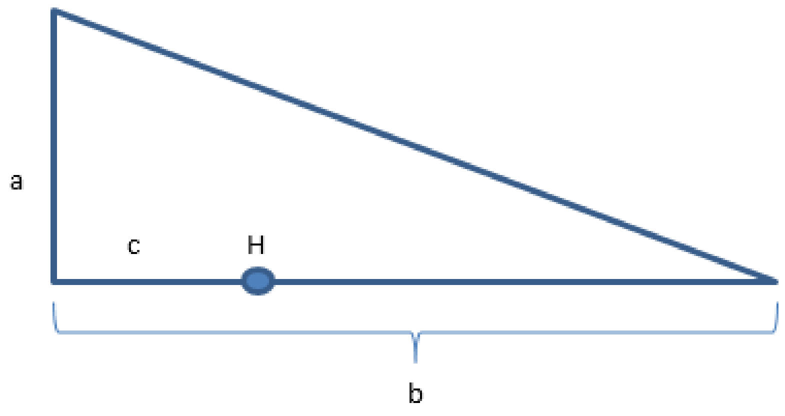
The planet Bliss has an ocean with a large number of triangular shaped islands. Each island is in the shape of a right triangle. The islands are of varying sizes, but the orientation of each island is such that the three vertices are located at the Northern point, the Southwestern point and the Southeastern point of the triangle. The right angle is located at the Southwestern point, which is due south of the Northern point and due west of the Southeastern point. The hypotenuse runs from the Northern point to the Southeastern point, and is of course the longest side of the island. The shortest side of an island may either be the side from Northern point to Southwestern point or from Southwestern point to Southeastern point.

On each island, the seat of government is the Custom House located at Northern point. But all the hotels are located along the southern shore between Southwestern and Southeastern points. The islands are extremely mountainous, and there is only one road on each island, which runs along the coastline around the entire island. There are no roads in the interior of any island. We would like to determine the shortest path to get from a hotel to the Custom House. There are only two realistic alternatives: should we start out going east (i.e. initially towards Southeastern point) or west (initially towards Southwestern point)?

The figure below illustrates the problem. In the figure, **H** represents the location of the hotel. An instance of the problem is defined by three numbers (**a**, **b**, and **c**) the represent the following distances in miles:

- The first number, **a**, is the length of the longitudinal side (i.e., the distance from the Northern point to the Southwestern point).
- The second number, **b**, is the length of the latitudinal side (i.e., the distance from Southwestern to Southeastern points).
- The third number, **c**, is the distance from the hotel to the Southwestern point.

For example, suppose these distances are 3, 4 and 1, respectively. If we leave the hotel going west, we will travel 1 mile to the Southwestern point, and then 3 miles to the Northern point. If we leave the hotel going east, we would travel 3 miles to the Southeastern point and then 5 miles to the Northern point along the hypotenuse. Since 1+3 < 3+5, we are better off starting to go west.



Throughout this problem, you may assume that the two distances will not be equal, so that one direction will definitely be less than the other.

Input

The input will be formatted as follows.

The first line will contain a positive integer n, $2 \leq n \leq 100$, indicating the number of islands. Each subsequent line of input will pertain to one island, and will contain the three positive integers representing an instance of the problem.

Output

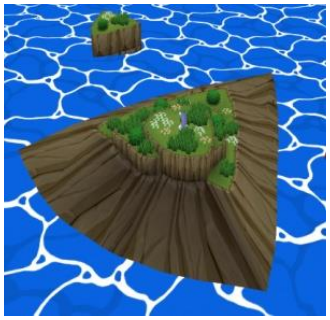
Your output should give the number of the test case followed by the phrase `go east` or `go west` as appropriate, as advice for how to leave the hotel to follow the shortest path to the Custom House.

Sample Input

```
3
6 8 1
6 8 7
3 3 2
```

Output Corresponding to Sample Input

```
1 go west
2 go east
3 go west
```



Problem 6

Whats a Hanin?

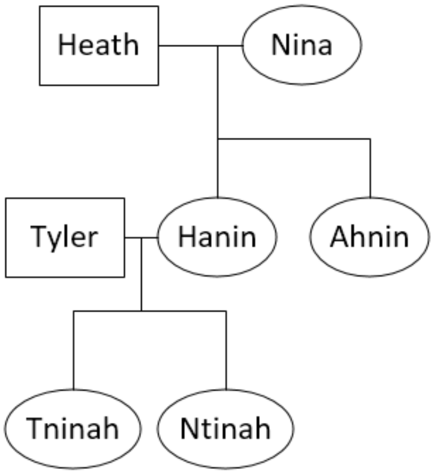
The Chewmen family has a long history of wacky names. This is because for as long as anyone can remember their first names follow these specific rules.

1. The Chewmen family only has female children.
2. Each daughter's name is formed by reversing the letters in the mother's name, and then placing this name following the first letter of the father's name.
3. The first sibling name takes the reversed mother's name, and places the first letter from the father's name at the (*n* mod *k)* index in the mother's name where *k* is the length of the mother's name.

For example, in 1750, the Chewmen family had two parents named Heath and Nina.

Heath and Nina had two daughters. The first was named **H**anin. Hanin's first sibling (*n = 1*) was named A**h**nin formed by placing the **h** into index 1 (1 mod 4) of the string "Anin" of length 4. If Heath and Nina had four daughters, they would both be named Hanin.

Assume all names start with a capital letter, and are followed by zero or more lowercase letters. The maximum name length is 100 characters, and a name never consists of any spaces or non-letters.



Input

The first line of input contains a single integer *c*, (1 ≤ *c* ≤ 1000), which is the number of test cases to follow. This is followed by *c* test cases. Each test case consists of a line with three names separated by a single blank space in this format: `mother father daughter`

Output

For each test case, output *g*, the number of generations where 1 ≤ *g* ≤ 100 that could link the two parents and a possible daughter. If there is no possible connection, output `No link`.

Sample Input

```
4
Nina Heath Hanin
Aubrey Ollie Yerhbuainxo
Catherine Phillip Cwpaathyietjrinel
Bonnie Jessie Einngobj
```

Output Corresponding to Sample Input

```
1
5
8
No link
```

Problem 7

Handshakes

Meetings between multiple groups can sometimes take a long time, mainly due to introductions. For instance, two groups of two people would require four handshakes, but three groups of two would require 12, since everyone from each group would need to shake the hands of each person from every other group. Given a limited amount of time, introductions alone can take up most of a meeting.

Your task is to calculate how many handshakes would be required in a meeting between multiple groups. For instance if three groups met, everyone from group 1 would have to shake hands with everyone in group 2 and group 3, and also everyone from group 2 would have to shake hands with everyone from group 3.

Input

The input will start with a single integer $n \geq 1$ indicating the number of groups that are meeting. This is followed by n lines, each with an integer on it denoting the group size. There will be at most 100,000 groups with at most 20 people in each.

Output

For your output, you only need to print the number of handshakes `mod 1000` it would require for every person from each group to meet every person from every other group.

Sample Input

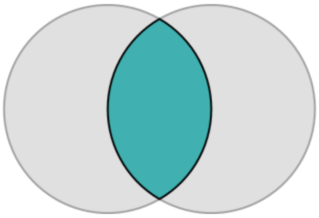
```
3
2
2
2
```

Output Corresponding to Sample Input

```
12
```

Problem 8

Overlap Length



Write a program that compares two strings, *string1* and *string2*, and calculates their overlap length. For example, the strings “FURMAN” and “MANKIND” have an overlap length of 3. Overlap length is the string of the longest length at the end of *string1* which can also be found starting at the beginning of *string2*. "APPLE” and “LEMON" have an overlap length of 2, while “LEMON” and “APPLE” have an overlap length of 0. "ALE" and “APPLE” would have an overlap length of 0, while “ALE” and “ALE” have an overlap length of 3.

Input

The input will start with a single line containing a positive integer *n* representing the specified number of pairs of lines to be input. Each pair of lines consists of two strings starting with a single character that is not a white space character (tab, blank, newline). This single character is always a letter, and will be followed by zero or more letters. The last character on each line is a newline character. All strings should be converted to uppercase upon input, and before comparison. Each string will consist of 1 to 80 letters.

Output

Format output as shown below where *string1* and *string2* both appear in uppercase with the string “` and `” between them, and the string “` have an overlap length of `” following. Note that *string1* is always the first string of the two strings to be input.

Sample Input

```
5
football
ball
ALE
Apple
Clemson
son
SC
C
APPLE
LEMON
```

Output Corresponding to Sample Input

```
FOOTBALL and BALL have an overlap length of 4
ALE and APPLE have an overlap length of 0
CLEMSON and SON have an overlap length of 3
SC and C have an overlap length of 1
APPLE and LEMON have an overlap length of 2
```