1: Hickory Dickory Dock

Description:

_Hickory dickory dock, the mouse ran up the clock._
_The clock struck one, the mouse “fell” down._
_Hickory dickory dock._

The poor mouse was apparently struck by the chime hammer in the clock. An unusual mouse trap indeed! Your task is to calculate how many mice could be struck in a given span of time. Consider that the hammer strikes on the hour for the given number of hours, one strike per second. It also strikes once on the quarter, half, and last quarter hours.

Input: There may be multiple time spans for consideration. The time span for each case will appear on a single line and consist of 6 integers. These represent the hour, minute, and second to begin the time, followed by the hour, minute, and second to end the time. The time span includes the starting time but not the ending time. Each time span will be for at least one second and for less than 12 hours. The integers will be delimited by spaces. For the VB programmers, commas will be used. The last line will consist of six 0’s. Assume all data is valid.

Output: For each time span use a separate line containing the message “1 mouse could be struck.” or “n mice could be struck.” where n will be the correct number. In the sample there are three cases. The first has the chime at 1:00:00. The second has the two chimes at 3:00:00 and 3:00:01, but not the chime at 3:00:02. The third has the chimes at 12:15:00, 12:30:00, 12:45:00, 1:00:00, and 1:15:00, but misses all the twelve chimes that go from 12:00:00 through 12:00:11.

Sample Input:

```
1 0 0 1 0 1
3 0 0 3 0 2
12 0 12 1 20 2
0 0 0 0 0 0
```

Sample Output:

```
1 mouse could be struck.
2 mice could be struck.
5 mice could be struck.
```
2: The Three Little Kittens

Description:

The three little kittens lost their mittens and they began to cry . . .

You get to help these poor kittens find their mittens! They lost them in a rectangular yard 5 units by 20 units in size. Each mitten is 2 units by 3 units with one corner unit missing (it represents the shorter thumb.) There are 6 mittens to find, two per kitten, each mitten marked with the kitten’s initials all over it.

Input: The names of the kittens are given on the first line. The names will all begin with different letters, and each will be no more than 20 letters in length. On the following lines is the yard represented by 5 lines of 20 letters. All input is delimited by spaces (plus commas for the VB programmers). The letters used as initials will NOT appear anywhere except on the mittens. No mittens overlap or are on edge, though some may be palm down and others palm up. Furthermore, the two mittens for a kitten will not be touching each other.

Output: For each kitten print its name followed by the coordinates of that kitten’s lost mittens (in either order). Let the upper left corner be (1,1) and the lower right corner be (5,20). Use the missing corner of the mitten as the location coordinates. Format as in the sample output, though you may space it more freely.

Sample Input:

Muffin Tigger Fluffy
A B C D E E G H I J K L L N O P Q R S S
A B M M E E G H F J K L L N O M M M S S
A B M M M E G H F F K L T N O P M M S S
F F C D T T G H F F K L T T O P Q R S S
F F F T T T G H I J K L T T O P Q R S S

Sample Output:

Muffin (2,5) (3,16)
Tigger (4,4) (3,14)
Fluffy (4,3) (2,10)
3: As I Was Going to St. Ives

Description:

As I was going to St. Ives,
I met a man with seven wives.
Each wife had seven sacks.
Each sack had seven cats.
Each cat had seven kits.
How many were there going to St. Ives?

We shall generalize this problem, assuming the traffic was merging so that all these characters were going to St. Ives! The characters might be different and the numbers might be different.

Input: The description of each character appears on a separate line beginning with the name and followed by how many of the next character it can hold. The last character can obviously hold only 0 of the next (non-existing) character. Assume that otherwise only positive integers are used and there will be no overflow. Names may be up to 20 letters in length. There will be at least one character and possibly many different characters.

Output: Indicate how many characters in all are going to St. Ives. Format as in the sample output.

Sample Input:

man 7
wife 7
sack 7
cat 7
kit 0

Sample Output:

There are 2801 characters going to St. Ives.
4: Jack and Jill

Description:

*Jack and Jill went up the hill*
*to fetch a pail of water.*
*Jack fell down and broke his crown.*

Well, well, well! Though it is curious that the well is at the top of a hill, let’s assume that the water was spilled right next to the well and starts flowing down the hill in the steepest direction. Your task is to determine where the water reaches the bottom of the hill.

**Input:** The hill is represented by an 6 by 6 matrix of integers representing elevations. The bottom of the hill is the outer edge of elevations. The top is somewhere inside, and has the highest elevation. Water does not flow diagonally. You may assume there is no ambiguity (only one correct direction exists from any point on the water’s path). Once it gets to the edge, the water is considered to be at the bottom of the hill, even though there may be lower elevations along the adjacent edge positions. There may be low spots on the hill such that the water never reaches the bottom. A level spot is NOT considered to be a drop. The elevations are delimited by spaces (or commas for VB programmers), each row on a separate line.

**Output:** On one line list the sequence of elevations that the water takes. On the following line give the message “The bottom is reached.” or the message “A pool is formed on the hill.” as appropriate.

**Sample Input:**

```
2 4 3 5 2 1
5 8 9 5 4 3
4 6 12 11 5 2
1 2 7 6 2 1
0 4 8 3 0 0
0 1 7 1 1 2
```

**Sample Output:**

```
12 6 2 1
The bottom is reached.
```
5: Twinkle Twinkle Little Star

Description:

Twinkle twinkle little star.
How I wonder what you are! . . .

The color and brightness of a star can be used to guess its type. Assume the following indicators:

• red and dim – distant red giant
• white and dim – near white dwarf or distant regular star
• red and bright – near red giant
• blue and bright – near very hot star
• blue and dim – distant very hot star
• yellow and bright – near sun-like star
• yellow and dim – distant sun-like star

You are to determine the type of a star given its color and brightness.

Input: The first line contains a positive integer indicating the number of stars to identify. Each following line contains the two word description of one star. The two words are delimited by a space (and a comma for VB programmers.)

Output: Indicate the types of stars, one per line. If there is no match, indicate “Unknown!”

Sample Input:

3
blue dim
dim blue
white bright

Sample Output:

distant very hot star
distant very hot star
Unknown!
6: Mary Had a Little Lamb

Description:

Mary had a little lamb. It’s fleece was white as snow.
And everywhere that Mary went, the lamb was sure to go. . . .

Apparently the lamb always stayed close to Mary – at school, at play, upstairs in bed, etc. Your task is to determine the furthest away the lamb ever gets from Mary.

Input: Locations are specified as three-dimension coordinates, all in feet. There will be multiple lines, one line for each observation time. Two locations are given for each observation time, the first for Mary, the second for the lamb. The last line will contain all 0’s. The (real) numbers will be delimited by spaces (and commas for VB programmers).

Output: Indicate the maximum distance in feet between Mary and the lamb. Format as in the sample output, though the spacing and number of decimal places may vary.

Sample Input:

3.0 10.0 20.0 3.0 10.0 22.0
2.5 25.2 100.8 2.5 30.4 125.0
1.5 69.3 26.2 10.2 60.0 30.1
0.0 0.0 0.0 0.0 0.0 0.0

Sample Output:

The maximum distance is 24.7524 feet.
7: Jack Be Nimble, Jack Be Quick

Description:

*Jack be nimble, Jack be quick.*
*Jack jump over the candle stick.* ... 

Jack was an ancestor of Evel Knievel. Jumping just one candle stick became boring, so he started lining many candlesticks in a row and jumping all of them. Given a parabolic line that Jack follows as he jumps, and that the candles are all the same length and are spaced evenly apart, you are to determine how many candles Jack can successfully jump. Keep in mind that Jack wants to stay above the flames, so you must allow for that height also.

Input: The input consists of five real numbers per line, each line representing one case. The first number is the height of the candles and will be greater than 0.0. The second number is the spacing between candles on the floor and will also be greater than 0.0. The last three numbers represent the coefficients of the parabola $Ax^2 + Bx + C$ respectively. All units are in inches. The numbers will be delimited by spaces (and commas for VB programmers.) The last line contains all 0’s.

Output: Allowing 2.0 inches for the flame in addition to the height of the candles, calculate the maximum number of candles Jack can jump. If the parabola opens upward, degenerates to a straight line, or never rises above the y axis, indicate with the message “Not a valid trajectory.” Consider the case of Jack’s height being exactly the same as the height of a candle being jumped (plus flame) as being permitted. Format as in the sample output.

Sample Input:

```
4.5 2.0 -1.0 100.0 0.0
4.5 2.0 1.0 100.0 0.0
0 0 0 0 0
```

Sample Output:

```
Jack can jump x candles.
Not a valid trajectory.
```
9: Picnic Items

Description: People are bringing things to a picnic. Generally there are multiple items, such as a dozen doughnuts, 24 cans of pop, etc. Your task is to determine exactly how many items are being brought in.

Input: Each line of input will contain information for two different things being brought to the picnic. These are as follows: the name of the item, how many of the item, the name of the next item, how many of the next item. The last line will have 0’s for the numbers of both items. The input values will be delimited by spaces (and commas for VB programmers).

Output: Echo the input (Be neat, printing in nice columns.) Leave a blank line. Then print the message ”There are n items at the picnic.” (where n is the total number of items.)

Sample Input:

Apples 24 Bananas 36
Cantaloupe 5 Doughnuts 24
xxx 0 xxx 0

Sample Output:

Apples    24    Bananas   36
Cantaloupe  5    Doughnuts  24

There are 89 items at the picnic.