

Inch By Inch
Input File: InchIn.txt

Breanne the grasshopper has decided to do some traveling. Being too young to drive and not a big fan of public transportation, she has decided to hop from one city to another. She always hops exactly two inches per hop. Your task is to determine the number of hops required to complete her journey. Note: there are 5,280 feet in a mile.

Inputs:

There will be one line of input that will contain the number of miles Breanne will travel on her journey.

Output:

There will be one line of output that contains the number of hops required to complete her journey, annotated exactly as shown below.

Sample inputs

125

Sample output

Breanne will make 3960000 hops

Change Input File: ChangeIn.txt

After counting up her receipts one day, Nora realized how her crasher *Knownomath* got his name. He can't do math, which means he has not been dispensing the proper change to her customers. Nora has decided to write a program that will calculate the correct change her customers' should receive, or inform them of the amount they have underpaid. A customer always purchases one or more instances of the same item.

Inputs:

The first line of input contains the number of customers to consider. This will be followed by one line of input per customer. Each line will contain an integer, the number of items purchased, followed by two doubles that are the cost of the item and the amount of money the customer gave to *Knownomath*. All inputs on a line will be separated by a space.

Outputs:

There will be one line of output per customer, which will contain either the change owed to the customer, or the amount the customer has underpaid. The output must be formatted and annotated as shown below.

Sample Input

```
3
2 0.1 1.0
20 30.59 500.67
300 30.59 100000.00
```

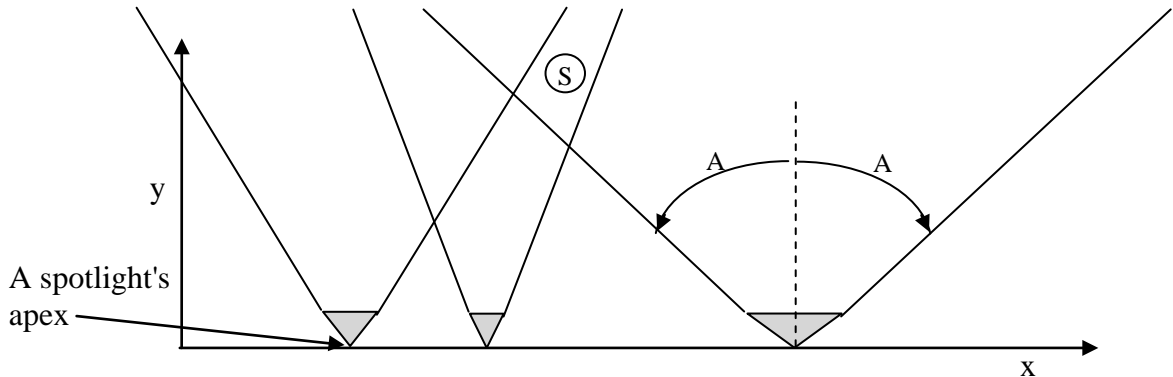
Sample Output

```
Your change is $0.80
Please pay and additional $111.13
Your change is $90,823.00
```

Spotlights

Input File: SpotsIn.txt

Evie owns a flat rectangular piece of property that has a set of north pointing spotlights installed ay around level along its south border line. The lights are used to illuminate the center of a south facing sign located on the property at ground level. Each light has its own focus angle, A , and its own intensity. The locations of the lights and the sign are expressed as Cartesian coordinates whose origin is the south-west corner of the property. A typical light set up is shown below as viewed from above the property.



The intensity of the light emitted from a spotlight at any point on the property is inversely proportional to the square of the point's distance from the apex of the spotlight, or the intensity is zero if the point is outside of the spotlight's focus angle. The apex of each spotlight is located on the x axis. Your task is to determine the total intensity of the light hitting the center of the sign. You need not consider sign's or the lights' height above the ground.

Inputs:

The first line of input contains the number of pieces of property to consider. This will be followed by two lines of input for each piece of property. The first line will contain three integers: the x coordinate of the center of the sign, followed by the y coordinate of the center of the sign, followed by the number of spotlights on the property line. The second line will contain three integers per spotlight: the x coordinate of the spotlight's apex, followed by the light's focus angle, followed by the intensity of the spotlight. All inputs on a line will be separated by a space.

Outputs:

There will be one line of output per piece of property, which will contain the total light intensity at the center of the sign, rounded to two digits of precision with a leading zero.

Sample inputs

```
4
10 5 1
6 45 16
5 5 3
0 50 100 5 45 100 10 40 100
```

10 2 2
10 15 400 40 5 500
10 4 1
10 45 16

Sample outputs

0.39

6.00

100.00

1.00

Square Sums
Input File: SquareIn.txt

Mathematician Ryan always liked producing numbers, and being a Texan he especially likes producing big numbers. Knowing this, you have decided to give him a program for his birthday that generates the value of the sum of the squares of the integers from 1 to n; $n \leq 10^9$. For example, for $n = 4$ the sum would be $1 + 4 + 9 + 16 = 30$.

Inputs:

The first line of input contains the number of values to compute. This will be followed a line that contains a the value of n to be used in each computation.

Output:

There will be one line of output per computation containing the computed value.

Sample inputs

3
4
3
123456

Sample outputs

30
14
627221052446816

Sudoku

Input File: SudokuIn.txt

A Sudoku puzzle consists of a rectangular grid of cells arranged in C columns and R rows. Within the grid the cells are grouped into rectangular groupings each of which are W cells in width and H cells in height. W is always a factor of C , and H is always a factor of R . Initially, the cells contain numbers in the range 0 to $W \cdot H$. During the solution to the puzzle, the zeros within the cells of a grouping are replaced with numbers in the range 1 to $W \cdot H$. Below is an initialized puzzle for $C = 8$, $R = 10$, $W = 2$, $H = 5$.

	0	1	2	3	4	5	6	7
0	8	4	0	4	8	8	1	2
1	1	2	0	5	0	0	0	6
2	0	6	0	0	8	4	1	5
3	0	7	0	6	0	0	6	0
4	1	1	8	0	8	8	0	0
5	0	1	0	7	0	4	7	9
6	0	0	0	0	0	0	4	7
7	0	9	4	0	0	7	7	7
8	0	4	0	0	2	0	9	9
9	0	4	0	0	2	0	4	7

Given a cell's column and row number, determine if there are any duplicate numbers in the cell's grouping, ignoring the zeros. If there are no duplicates output zero. Otherwise output the total number of reoccurrences of the numbers within the cell's grouping, ignoring zeroes and not counting the original occurrences. For example, given row 9 and column 6 in the above puzzle, the output should be 7.

Inputs

The first line of input contains the number of puzzles to consider. This will be followed by one data set for each puzzle. The first line of a data set contains the number of columns and rows (C and R) in the puzzle, followed by the width and height (W and H) of the cell groupings. Each of the next R lines of a puzzle's data set contains W integers, that are the values contained in one row of the puzzle beginning with row 0 and ending with row $R-1$. The last line of a data set will contain an integer, n , followed by the column and row numbers of n cells in the puzzle. The row and column numbers start from zero, and all inputs on a line are separated by a space.

Outputs

For each puzzle there will be n lines of output, one line for each cell whose row and column number was specified on the last line of the puzzle's data set. If there are no duplicate

numbers in that row and column number's W by H grouping output zero, otherwise output the total number of reoccurrences of the numbers within the cell's grouping, ignoring zeroes and not counting the original occurrence.

Sample inputs

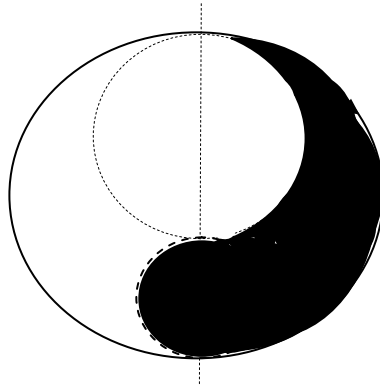
```
2
8 10 2 5
8 4 0 4 8 8 1 2
1 2 0 5 0 0 0 6
0 6 0 0 8 4 1 5
0 7 0 6 0 0 6 0
1 1 8 0 8 8 0 0
0 1 0 7 0 4 7 9
0 0 0 0 0 0 4 7
0 9 4 0 0 7 7 7
0 4 0 0 2 0 9 9
0 4 0 0 2 0 4 7
4 7 7 0 0 3 2 4 4
12 8 3 8
8 2 0 4 0 0 1 2 3 0 0 0
1 2 0 5 8 0 0 6 0 1 1 3
0 6 0 4 0 4 0 5 0 8 0 0
0 7 0 6 0 0 0 0 0 0 6 0
0 0 0 0 0 8 0 0 4 2 0 5
0 1 0 0 0 4 9 8 0 4 9 7
0 0 0 0 0 0 0 0 0 0 0 0
0 9 4 0 0 7 0 7 0 0 0 1
4 0 7 6 7 11 7 5 2
```

Sample output

```
7
2
0
4
2
0
2
4
```

YinYang
Input File: YinYangIn.txt

The symbol shown below approximates the Chinese duality symbol Taijitu. It is comprised of an outer circle, which contains an upper and a lower inner circle inside of it. The vertical diameters of the three circles are collinear. The inner circles are tangent to each other and they are tangent to the outer circle. The black area of the symbol is call yin, and the white area is call yang. Given the radii of the outer circle and the upper-inner circle, calculate the area of the yin (black) and yang (white) areas of the symbol. Use 3.141592653589793 for the value of pi.



Inputs

The first line of input contains the number of Taijitu symbols to consider. This will be followed by one line of data per symbol that contains two integers: the radius of the outer circle followed by the radius of the upper-inner circle. The radii will be separated by one space.

Outputs

There will be one output per line that contains the yin area of the symbol followed by the yang area of the symbol each to three digits of precision. The outputs on a line should be separated by a space.

Sample inputs

```
2
10 6
12 2
```

Sample output

```
125.664 188.496
376.991 75.398
```


Golf

Input File: GolfIn.txt

Logan is a very good golfer who always hits the ball in a straight line directly at the hole. When struck, the ball always travels the exact distance that is associated with the club he used. Logan never selects a club that will cause the ball to go beyond the hole. Your task is to determine the minimum number of strokes Logan will take to complete each hole of his golf round, given the distances associated with each club in his bag, and the distance from the tee box (starting point) to the hole. Any club can be used more than once to complete a golf hole.

Inputs:

The first line of input contains the number of golf holes to consider. This will be followed by two lines of input per hole. The first line will contain two integers: the distance to the hole and the number of clubs in the golf bag Logan will carry with him on that hole. The second line will contain the distance associated with each club in the bag. All inputs on a line will be separated by a space.

Outputs:

There will be one line of output per hole, which will contain either the *minimum* number of strokes Logan will take to advance the ball the exact distance to the hole, or the phrase *No can do* if the clubs in his bag can not be used to complete the hole. Remember, Logan will never hit the ball beyond the hole.

Sample inputs

```
4
70 3
25 30 10
321 6
25 50 10 15 90 200
320 6
25 50 10 15 90 200
230 3
25 30 10
```

Sample output

```
3
No can do
4
8
```

Folding
Input File: FoldingIn.txt

Being tired of trying to jam as many items of clothing into her suitcase as possible. Maggie has purchased a newly invented digital suitcase to take with her on vacation. The suitcase accepts the clothing items in the form of a string of ASCII characters, S , folds them into a single signed integer DS , and then the integer is placed into the traveler's carry-on bag.

The folding process divides the ASCII characters of the string S into groupings of four sequential characters beginning with the first four characters in the string. Then these groupings are added to the integer DS one grouping at a time, ignoring overflow. If the last grouping does not contain four characters, the grouping is right justified and padded with zeros on the left to complete the 32 bit grouping. Spaces are considered ASCII characters.

For example, if Maggie was taking the following clothing items on vacation: *3 Pants*, the addition process would be as shown below. Note: the left most eight bits of the second grouping have been set to zero, because the grouping only contains three characters.

$$\begin{array}{r}
 3 \text{ Pa} \quad 00110011001000000101000001100001 \\
 + \text{nts} \quad + 00000000011011100111010001110011 \\
 \hline
 \text{xxxx} \quad 00110011100011101100010011010100 = DS = 864994516_{10}
 \end{array}$$

If she were taking *2 Hats 3 Caps* the addition process would be as shown below, where $zzzz$ and $yyyy$ represent the intermediate subtotals:

$$\begin{array}{r}
 2 \text{ Ha} \\
 + \text{ts } 3 \\
 \hline
 zzzz \\
 + \text{Cap} \\
 \hline
 yyyy \\
 + \text{ } s \\
 \hline
 \text{xxxx} = DS
 \end{array}$$

Inputs:

The first line of input contains the number of vacations to consider. This will be followed by one line of input per vacation that describes the clothing she will taking with her on that vacation.

Outputs:

There will be one line of output per vacation that contains the signed integer she places into her carry-on bag.

Sample Input

4
3 Pants

2 Hats 3 Caps
3 Pants 2 shorts 4 Coats
Caps

Sample Output

864994516
-959001993
-759357422
1130459251

