Problem Set

Please check that you have **10** problems and **15** sheets.

Problem	Problem Name	Balloon Color
A	Spinster	Yellow
B	Selfie Selfie Star	Lime
C	Picking Blueberries	Blue
D	Car Showroom	Green
E	Help Decryption	Pink
F	Product Distribution	White
G	Concrete Pathway around Pool	Red
H	Counting Vertex	Orange
I	Money Drop Game	Silver
J	Numbers Conversion	Purple

Note: The input and output for all the problems are standard input and output.

October 26, 2017

Problem A: Spinster

Miss Spinster is a professor from Medical Institute. At the end of the academic year, she wants to take her students on an excursion to the upper part of the country. She would like to go along with all of her students but she is afraid that some of them might become couples during the excursion period. She wants to prevent her students from this situation. So, she defined the following rules that indicate a low probability two persons will become a couple:

- Their height differs by more than 40 cm.
- They are of the same sex.
- They have the aim to become different type of medical specialist.
- They have an opposite interest in performing arts.
- Their interest in leisure activity is different.

So, for any two persons that she brings on the excursion, they must satisfy at least one of the requirements above. Your job is to help Miss Spinster by writing the program that find the maximum number of students that she can take on her excursion.

Input

The first line of the input consists of an integer $T \leq 50$ giving the number of test cases. The first line of each test case consists of an integer $N \leq 50$ giving the number of pupils. Next there will be one line for each pupil consisting of four space separated data items:

- an integer h giving the height in cm;
- a character 'F' for female or 'M' for male;
- a lower-case string describing the type of medical specialist;
- a character 'Y' for interest in performing arts or 'N' for not interest;
- a lower-case string with the name of the leisure activity.

No string in the input will contain more than 50 characters, nor will any string contain any whitespace.

Output

For each test case the following output will be generated.

Sample Input	Sample Output
2 4 35 M Cardiologist Y Reading 40 M Immunologist Y Sport 30 F Cardiologist Y Music 43 M Audiologist Y Music 9 27 M Pediatrician Y Reading 47 M Neurologist Y Music 150 F Pediatrician Y Music 51 M Immunologist N Sport 40 M Immunologist N Game 44 F Neurologist Y Shopping 34 F Immunologist Y Sport 35 M Neurologist Y Reading 39 F Cardiologist Y Reading	3 8

Problem B: Selfie Selfie Star

Facebook admin announces a contest to choose a selfie star in next week. Most of the girls in our university are interested in it. They would like to know which would be the most beautiful post taken by which angle position of the camera. Some selfie analyzers say that 45-degree position of the selfie camera is the best position to get a beauty post. The girls start to analyze and take many selfie photos in different positions of the camera.
Can you help them to find out the best posts, which will be appeared in the camera to be a selfile star?

Input

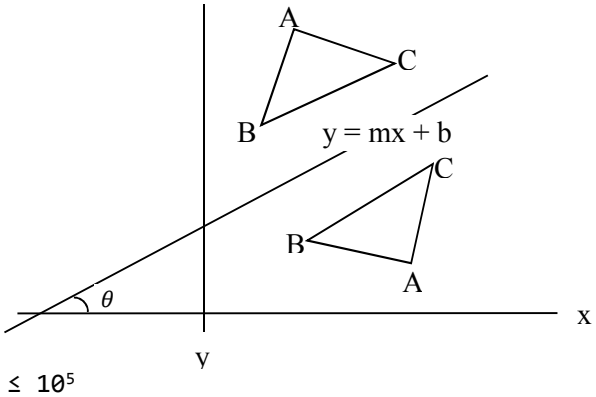
- First line contains **T**, the number of test cases.
- The second line shows the space-separated string and integer representing the original view of each girl, G and the number of vertices of the girl **V**.
- Third line shows the position of the camera with two separated strings: m and b where m is the slope of the camera and b is the y intercept in the slope intercept equation

  $$y = mx + b.$$

- This is followed by V lines, each containing the two space separated integers **x** and **y** coordinates for the original view of girl.

Output

Print the respective x and y coordinates of each girl, which will be appeared in the reflected selfie camera as shown in the figure.



Constraints

- $1 \le T \le 10^5$
- $1 \le G \le 10^5$
- $3 \le V \le 30$

Sample Input	Sample output
2 G1 4 0.5 2 2 4 4 6 2 6 3 5.5 G2 3 -1 0 2 4 4 6 2 6	G1 4 6.00 -4.00 8.80 -3.60 7.60 -5.20 7.80 -4.10 G2 3 -4.00 -2.00 -6.00 -4.00 -6.00 -2.00

Problem C: Picking Blueberries

Thuzar was assigned by her mom to pick up the blueberries in such a way that she may not exceed the limit proposed. Harvesting blueberries is a simple process but, even so, there are a few things to keep in mind.

When picking the blueberries, she noticed that if she pick from the bush i, she couldn't pick the blueberries at the bush i+1 (some sort of magic in their land).

Worried about this, Thuzar wants to know the maximum blueberries she can pick, given the number of bushes and the number of blueberries in each bush.

So, you are supposed to help her by writing a program. Your program finds the maximum blueberries with constraints of not being able to pick adjacent bushes and not more than the limit proposed.

Input

The first line of the input contains a single integer T, (1<=T<=10) that indicates the number of test cases. Each test case starts with a number N and K, being N the number of bushes and K the number of blueberries Thuzar will pick as maximum, the next line contains N integers, each one representing the number of blueberries in each bush.

Output

For each test case, output the maximum number of blueberries Thuzar can pick. See the samples for the exact output format.

Sample Input	Sample Output
5 5 100 50 10 20 30 40 4 55 70 40 50 5 4 42 13 28 25 15 2 100 110 150 5 50 10 20 30 40 60	90 50 38 0 50

Problem D: Car Showroom

The XYZ car showroom located in downtown displays various types of cars. The manager of the showroom decided a strategy to show cars for customer satisfaction because of the limited space of the showroom. When a customer requests for a type of car and this type of car is not there, the manager takes the car to the space (empty place) in the showroom. However there are limited space to show the cars and when the space is not empty, the manager replaces the least recently showed car with the current requested car. Your task is to compute the number of car placement in the showroom the manager has performed when customers request the various cars.

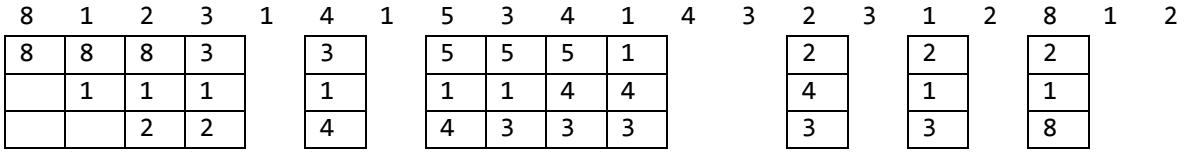


Figure 1. Illustration of Car Placement

Example: Let's consider the example on Figure 1. The types of car requested by customer are: `8 1 2 3 1 4 1 5 3 4 1 4 3 2 3 1 2 8 1 2`. The showroom has three places to show the cars. The requests of first three customers make placement which fills the three empty places. The customer request to car type 3 replaces car type 8 because car type 8 was showed least recently, where as the most recently showed car type is car type 2. The request to car type4 replaces car type2, as car type2 was showed least recently and car type1 is the most recently showed car type. When the request to car type5 occurs, car type3 was showed least recently. The most recently showed car type is type1, and just before that car type4 was showed. Thus the manager replaces car type3, not knowing that car type3 is about to be requested. When the customer asks for car type 3, the manager replaces car type 4 since, of the three cars in places {1, 4, and 5}, car type 4 is the least recently showed. In this way, there are 12 car placement in the showroom.

Input

The first line of the input contains a single integer `T, (1<=T<=20)` that indicates the number of test cases. The following lines describe the datasets. Each dataset contains the following information:

- The first line contains integer `N`, where `N (3 ≤ N ≤ 10)` is the number of available places in the showroom.
- The second line contains integer `L`, where `L (N < L ≤ 10^5)` is the length of the customer request string.
- The third line contains L string s, where s is the request string of integer t separated by one space, where `t (1 ≤ t ≤ 100)` is the car type.

Consecutive car types in the request string are not the same.

Output

For each data set, write in one line the number of car placement.

Sample Input 1	Sample Output 1
3 3 20 1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6 4 22 1 2 3 4 5 3 4 1 6 7 8 7 8 9 7 8 9 5 4 5 4 2 3 20 8 1 2 3 1 4 1 5 3 4 1 4 3 2 3 1 2 8 1 2	15 13 12

Problem E: Help Decryption

Alice is interested in Data encryption. He encrypted a secret message and sent the encrypted message to Bob. His encryption uses an invertible matrix according to the following procedure. First, he assigns a number to each letter in the alphabet (with 0 assigned to a blank space, 1 to a, 2 to b, …, 26 to z). Then convert the message (including blank spaces) to numbers and partition it into row matrices, each row having *n* entries. If $X=[x_1\ x_2\ \dots\ x_n]$ is a uncoded $1\times n$ matrix and A is an invertible $n\times n$ matrix, then Y=XA is the corresponding encoded matrix.

For example, let the message be 'still alive', n=3 and $A=\begin{bmatrix}1 & -2 & 2\\ -1 & 1 & 3\\ 1 & -1 & -4\end{bmatrix}$.

Firstly, 'still alive' is converted to `19 20 9 12 12 0 1 12 9 22 5` and it is partitioned into row matrices with size 3. Therefore, the row matrices are `[19 20 9], [12 12 0], [1 12 9]` and `[22 5 0].` Since the number of letters in the original message is 11, one blank space is added in last row vector.

Then calculate for Y=XA as follows.

$$[19\ 20\ 9]\begin{bmatrix}1 & -2 & 2\\ -1 & 1 & 3\\ 1 & -1 & -4\end{bmatrix}=[8\ -27\ 62]$$

$$[12\ 12\ 0]\begin{bmatrix}1 & -2 & 2\\ -1 & 1 & 3\\ 1 & -1 & -4\end{bmatrix}=[0\ -12\ 60]$$

$$[1\ 12\ 9]\begin{bmatrix}1 & -2 & 2\\ -1 & 1 & 3\\ 1 & -1 & -4\end{bmatrix}=[-2\ 1\ 2]$$

$$[22\ 5\ 0]\begin{bmatrix}1 & -2 & 2\\ -1 & 1 & 3\\ 1 & -1 & -4\end{bmatrix}=[17\ -39\ 59]$$

Then he sent the encrypted message: `8 -27 62 0 -12 60 -2 1 2 17 -39 59`, to Bob.
Bob received the encrypted message and he also knows the number of letters in each row matrix, *n*, and the invertible matrix, *A*. But it is difficult to decrypt the message because of less knowledge in matrix algebra.

Your work is to help decrypt the message for Bob.

Input

The first line of the input contains a single integer T, (1<=T<=20) consisting of T test cases. Each test case starts with a line containing a string that is the encrypted numbers separated by a single space. Next line contains an integer, *n*, that gives the number of entries in a row matrix. And then to input the entries of an invertible matrix, A, follow *n* lines, each contains *n* integers separated by a single space.

Output

Display the original message.

Sample input 1	Sample output 1
2 8 -27 62 0 -12 60 -2 1 2 17 -39 59 3 1 -2 2 -1 1 3 1 -1 -4 32 55 38 67 2 2 4 2 1	still alive mcpc

Problem F: Product Distribution

The ICARE Company has *m* plants, (1, 2, …, m), located throughout a state and each plant has different capacities to supply. Each day the firm must furnish its *n* retail shops, (1, 2, …, n), and each has different demands. The transportation cost (in $) per item from plant *i* to retail shop *j* is c_{ij} in $. The production manager is planning to distribute the products satisfying the capacity that can supply from each plant and demand from all retail shops with a minimal transportation cost. Therefore, he considers applying minimum cost rule for distribution. This rule is:

- First select a pair of a plant and a retail shop with minimum transportation cost, and allocate as much to this retail shop as possible, but within the supply and demand constraints.
- Next, select another pair of a plant and a retail shop with minimum transportation cost and continue the above procedure until all of the supply and demand requirements are satisfied. In a case of tied minimum transportation cost between two or more distributions, the tie can be broken by selecting the retail shop that can accommodate the greater quantity. If tie still occurs (i.e., there will be equal quantity), select smaller plant number first. If the plant numbers are same, select smaller retail shop first.

You need to help the manager to determine how much should be shipped from each plant to each of retail shops to satisfy all of the supply and demand requirements with the above rule.

Input

The first line of the input contains a single integer T, (1<=T<=20) consisting of T test cases. Each test case starts with a line containing two integers, m and n, specifying the number of plants and the number of retail shops respectively.

This is followed by two lines, the first line containing m integers that give the supply from each plant and the second line containing n integers that give the demand from retail shop.

Then follow *m* lines, each contains *n* integers. The value at j^{th} column of i^{th} row specifies the transportation cost, c_{ij}, between plant *i* and retail shop *j*.

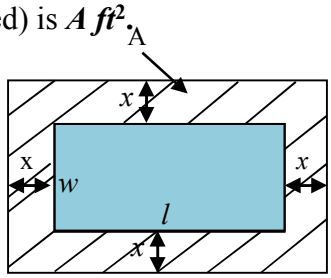
Output

For each test case, display all distributions from each plant to each retail shop (*m* lines with *n* columns) and total shipping cost. Output format: Case<*single space*><test case no.>:<output>

Sample input	Sample output
3 2 2 30 30 20 40 6 2 4 3 3 4 40 30 50 50 10 40 20 7 2 5 6 3 5 4 2 4 6 3 5 2 4 50 40 30 30 20 10 7 5 9 11 4 3 8 6	Case 1: 0,30, 20,10, 170 Case 2: 30,10,0,0, 10,0,0,20, 10,0,40,0, 460 Case 3: 20,0,20,10, 10,30,0,0, 560

Problem G: Concrete Pathway around Pool

Your company is going to build concrete pathway around the swimming pool. The prototype of the pool is shown in figure. The length and width of the pool are *l* ft and *w* ft, and the area of the pathway (single line shaded) is ***A ft²***.



The problem is to determine what will be the width ***x*** (in ft) of the pathway should be after the length, width and the area of pathway are already defined.

Input

The first line of the input contains a single integer T, (1<=T<=20) consisting of T test cases. For each test case, first line consists of two integers representing length, *l* and width, *w*, separated by single space. And next line consists of a positive number, *A*, that represents the area of the pathway around the pool.

Output

For each data set, print the width of the lane, x in two-decimal place.

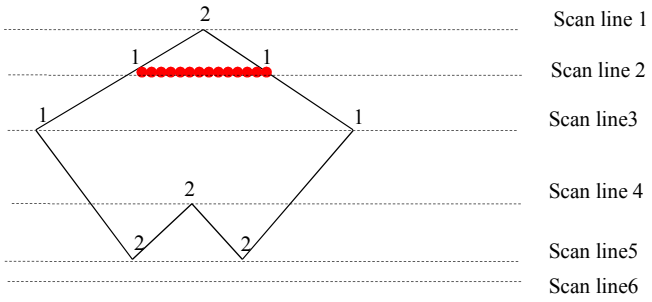
Sample input	Sample output
2 18 9 124 36 18 393.25	2.00 3.25

Problem H: Counting Vertex

A polygon is a shape formed by connecting line segments end to end, creating closed path. We can paint the area inside a polygon using a specified color using polygon fill method. The polygon fill method simply finding the intersection points for each scan-line with polygon edges and fill between alternate pairs of intersection points as shown in figure(for scan line 2). A scan line is a horizontal line from left to right.

But scan-line intersection at polygon vertices needs special handling. A scan-line passing through a vertex, intersection points for that vertex is counted as two (for scan line 1, 4, 5, 6) because two intersecting edges are both either above or below the scan-line. But it is not always true that the two intersecting edges sharing a vertex are on opposite sides of a scan-line (scan line 3), count the vertex as a single intersection point.

Your task is to write a program that counts the intersection points on each vertex of the polygon.



Input

The input starts with T (1<=T <=20) test cases which consists of the following lines:

- The first line containing a positive integer ***N***, the number of vertices of the polygon and 3<=N<=100.
- The following *N* lines containing a pair of Cartesian coordinates(x, y) in the *XY* plane, separated by one blank space. Every coordinate is an integer number where 1<=x<=640 and 1<=y<=480. These are the vertices of the polygon. The vertices of a polygon appear in clockwise order.

Output

The output would be the number of intersection points for each vertex (in clockwise order) in a space separated line.

Sample input	Sample output
2 7 20 10 20 26 30 50 40 40 30 20 25 25 45 15 5 20 10 0 40 30 80 60 60 30 50	Case 1: 2 1 2 1 2 2 1 Case 2: 2 1 2 1 1

Problem I: Money Drop Game

The money drop game is presented in a TV program. The young competitors interest this program and most of them played in this game. Some of them won little money from this program. The money drop program offers to the competitor fifty bundles of money firstly. In each bundle contains five lakh kyats. In each program, it is need to play two challengers. They need to guess the correct answer for eight questions. First four questions have four answers and they can guess only three answers. Second three questions also have three answers and they need to guess only two answers. Finally, they must guess only one correct answer out of two. They must bank up their money upon their answers. If the answer is correct, the money will remain upon the answer box. If it isn't correct, the money will drop to the box. After they have guessed all of questions, the remaining bundles of money are became their own. If all of money will be lost in each question, they must return back. Mr. Win (player name) interested this program and he joined this with his friend. They have several knowledge and they can always guess only impossible (incorrect) answer in each questions. They can definitely know the correct answer in some question. They must bank up all the remaining bundles to this answer in such questions. They will guess some answer if they will not know which answer is correct? For example, they will guess first question and they will bank up two-third of bundles in most likely possible answer. And then, they will guess with two-third of bundles in second possible answer and they will bank up the remaining bundles to the third possible answer. If the amount of two-third bundle will be between one and two, they will bank up the two bundles. How money they will get after finished the game?

Input

The first line of the input contains a single integer T, (1<=T<=20) consisting of T test cases. Each test case starts with a line containing an integer which shows the input lines to be followed. Each input can be pair of character for answer in first four questions. For example, the input can be `A1 B2 C3 DI A`. There are four possible answers such as `A, B, C`, and `D` in the first four questions; `A, B` and `C` in the second three questions; `A` and `B` in the last questions. `A1` means 1 is the most possible answer (`A`) in the question, `B2` means `2` is the second possible answer, `3` means the third possible answer and `I` means the impossible answer. If they will guess definitely correct answer, the input can be `A0 BI CI DI A`. `A0` means `0` is the definitely correctly answer (`A`) and; `BI, CI` and `DI` are not possible answer. The last character (`A`) represents the correct answer stored in the computer. The priority of the possible answer and the correct answer can be different in all questions.

Output

If they will be dropped their all bundles of money, the output will be sorry. If not, the amount of money will be shown that they owned. Output format : Case<*single space*><test case no.>:<output>

Sample input	Sample output
2 8 A0 BI CI DI A AI BI C0 DI C AI BI CI D0 D A1 B2 C3 DI A A1 BI C2 C A2 B1 CI B AI B2 C1 C A0 BI A 3 A1 BI C2 D3 A A1 B3 C2 DI B A3 B1 C2 DI A	Case 1:3000000 Case 2:Sorry

Problem J: Numbers Conversion

Roman numerals are an ancient number system but still have uses in the modern world. The Roman system used a series of letters to represent numbers. The key letters were as follows:

Letter	value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

The numerals can be combined to create all of the numbers between one and four thousand using a set of simple rules.

- The first rule is that the letters can be repeated several times, with the values of each being additive. This means that `I` is one, `II` means two and `III` is three. However, to avoid four same letters being repeated in succession (such as `IIII` or `XXXX`), the third rule is used.
- The second rule states that the larger numerals must be placed to the left of the smaller numerals to continue the additive combination. So `VI` equals six and `MDCLXI` is `1,661`.
- The third rule allows for a small-value numeral to be placed to the left of a larger value. Where this occurs, for example `IX`, the smaller numeral is subtracted from the larger. This means that `IX` is nine and `IV` is four. The subtracted digit must be at least one tenth of the value of the larger numeral. Accordingly, ninety-nine is not `IC` but rather `XCIX`. The `XC` part represents ninety and the `IX` adds the nine.

You are supposed to write a program that convert the decimal numbers to Roman numerals and vice versa.

Input

The first line contains integer T(1<=T<=10) which is the number of test cases. The input for each test case is given in a single line. This line contains a character ‘D’ or ‘R’, then an integer N and next N strings. Character ‘D’ or ‘R’, integer N and N strings are separated by a space. Here, character ‘D’ means conversion of Decimal number to Roman and ‘R’ means conversion of Roman numbers to Decimal. N is the number of Roman/Decimal values and N strings represent the values of Roman/Decimal numbers according to character ‘R’ or ‘D’. The value of Decimal/Roman numbers must be between 1 and 3999 inclusive.

Output

For each line of input number values, produce one line of output values. It contains the equivalent Decimal/Roman values for given Roman/Decimal numbers. The number values are separated by a space. See the samples for the exact format of output.

Sample Input	Sample Output
8 R 3 LXIX LXXXVII MMDCXXXV D 5 45 67 899 71 30 D 3 1661 3599 877 D 2 99 125 R 3 MMDLXXVIII MMCMXCIX DCCIII R 2 LXVII MCMLXXXIX R 4 XCIX MDCCVIII CCVIII LXXX D 2 999 98	69 87 2635 XLV LXVII DCCCXCIX LXXI XXX MDCLXI MMMDXCIX DCCCLXXVII XCIX CXXV 2578 2999 703 67 1989 99 1708 208 80 CMXCIX XCVIII