

Manitoba High School Programming Contest

University of Manitoba

May 25, 2012 12:30-3:30 PM

Problem Packs

Notes:

- All input should be read from standard input (the keyboard) and written to standard output (the screen).
- No text boxes or input windows should be used.
- Any programming language resources are allowed.

Problem 1 – Loan Payment

If you take out a loan for a fixed amount of time, what should your payments be? The answer is complicated by the fact that you are charged interest, so you can't just divide the amount of the loan to be repaid over the amount of time.

Fortunately, there's a formula for calculating the payment on a loan. Suppose you have a loan for N months, and the amount you have loaned is P dollars. Also, assume that the interest rate is r (as a decimal, i.e., 5 % would be $r = 0.05$). Then the monthly payment on your loan is

$$P*r*(1+r)^N / ((1+r)^N - 1)$$

Write a program that accepts three values, P , N and r , and then outputs the amount to be paid each month. The first two values are integers (no decimal point) greater than zero, and the last is a decimal number greater than 0 and less than 1.

Input

Your program should accept three quantities on separate lines of input: first the initial amount loaned (P), then the number of months of the loan (N), and then the interest rate as a decimal (r).

Output

Your program should output the integers payment in this format:

“The monthly payment is \$X. “

where X is a decimal number. Do not worry about rounding the output value.

Sample Input	Sample Output
20000 60 0.069	The monthly payment is \$1405.657672350561

Judging Data for Problem 1

Note: since this problem does not involve any looping, your program will be run separately on each of the four input cases listed below.

Case 1:

20000

60

0.069

Case 2:

100000

1

0.01

Case 3:

100000

100

0.001

Case 4:

1000

15

0.055564

Problem 2 – Cooling Pies

You are in the business of cooling pies. You are not in charge of baking them. Only cooling them.

You have determined that the **rate** at which a pie cools depends on three things:

1. the current temperature of the pie (T_p).
2. the current temperature of the room that the pie is being cooled in (T_r).
3. k , a constant, which depends on several factors, including the size of the pie, the type of pie shell used, and (you *think*) the type of filling. k is a **negative** number.

In particular, you notice that in one minute, the temperature of your pie will go down by $k(T_p - T_r)$ degrees (notice that $k(T_p - T_r) < 0$). However, the temperature of your room is not stable. Initially, it is T_0 degrees and increases by Q degrees each minute: after one minute the temperature of the room will be $T_0 + Q$ degrees, after two minutes it is $T_0 + 2*Q$.

Write a program that will calculate the temperature of the pie after a given number of minutes.

Input

The first line of input is a single number n that gives the number of test cases you have to process. Then for each case, you have 5 values on one line: T_p (an integer), T_r (an integer), k (a negative decimal), Q (an integer) and m (the number of minutes of cooling, an integer).

Output

For each test case, output "Case #X" where X is a number starting at one, then output the final temperature of the pie after m minutes.

Sample Input	Sample Output
3	Case #1 421.35
425 60 -0.01 3 1	Case #2 417.7665
425 60 -0.01 3 2	Case #3 414.248835
425 60 -0.01 3 3	

Judging Data for Problem 2

15

425 60 -0.01 3 1

425 60 -0.01 3 2

425 60 -0.01 3 3

425 60 -0.99 0 0

425 60 -0.99 3 0

350 68 -0.01 2 3

375 0 -0.05 1 15

400 -15 -0.3 4 60

400 0 -0.05 0 100

400 0 -0.05 0 0

1000 20 -0.003 0 300

1000 20 -0.002 0 3000

1000 20 -0.001 0 20000

1000 1 -0.001 0 100000

1000 -1000 -0.001 1 2000

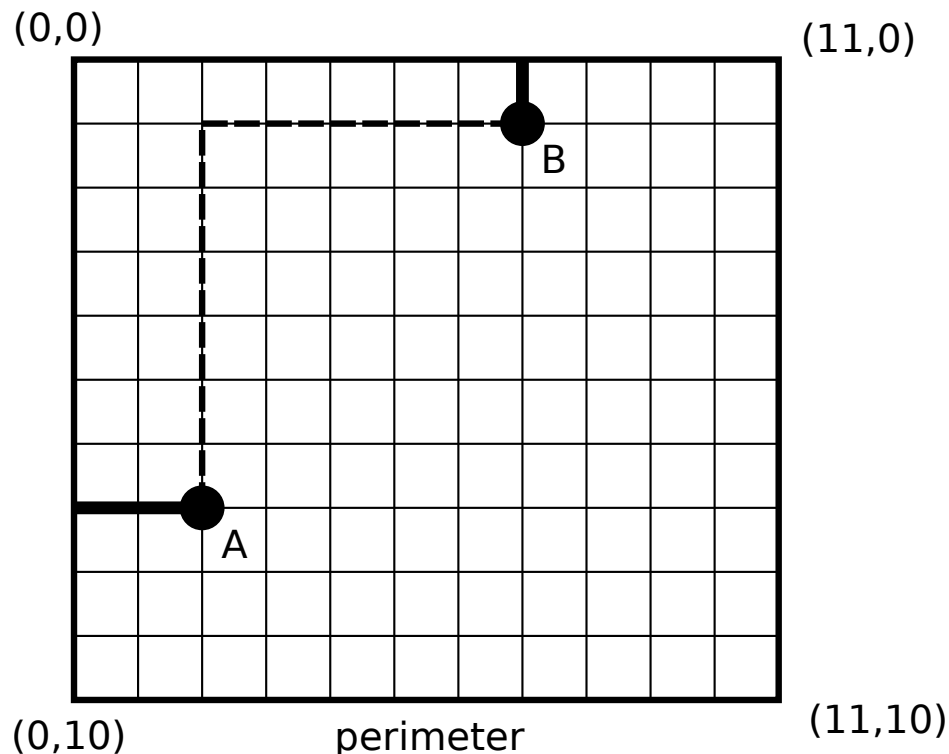
Problem 3 – The Winnipeg Distance

You may know about the "Manhattan Distance", which is a way of calculating the distance between two points in two-dimensional space. For points $p = (x_1, y_1)$ and $q = (x_2, y_2)$, their Manhattan Distance is $|x_2 - x_1| + |y_2 - y_1|$. (Here $|x|$ represents the absolute value of the number x , which is the value of x without the sign, e.g., $|-2| = |2| = 2$.) This measurement is called the Manhattan Distance because it represents the number of blocks you must drive to get between two spots when you drive along a road system laid out as a grid.

But Manhattan is an island, while Winnipeg is clearly not. So we propose the "Winnipeg Distance" for points in a two-dimensional region. Informally, the Winnipeg Distance will be just like the Manhattan Distance, but with the possibility of "taking the perimeter" if both points are near the boundary of the city. If you take the perimeter, then you simply add the distance from your starting point to the nearest point on the perimeter and the distance from your ending point to its nearest point on the perimeter. (The distance travelled on the perimeter does not count towards the distance, since that's just highway driving.)

The layout of Winnipeg in this example will be a rectangle of grid points where the exterior of the grid is the perimeter. Every road intersects with the perimeter at the edge of the grid (i.e., every road is connected to the perimeter twice).

Here is an example:



First notice that the point (0,0) is the upper left corner and the point (11,10) is the lower right corner. The x coordinates increase from left to right and the y coordinates increase from top to bottom.

Suppose we want to calculate the distance from A to B. A is at point (2,7) and B is at point (7,1). The Manhattan distance from A to B is $|2-7|+|7-1| = 11$. This is represented by the dotted line between A and B. But the Winnipeg distance is represented by the dark line that goes from A to the perimeter (at point (0,7)) and then from the point (7,0) to B. Thus the Winnipeg distance from A to B is 3, since the dark solid line is less than the dotted line.

Notice that the closest point to on the perimeter to a given point may be in any of four directions: you could reach the perimeter by travelling east, north, south or west, and your algorithm will need to check all four options.

Input

The first line of the file is the number of input test cases to examine. Then each line is given by six integers. First are the dimensions of the city, given by the maximum x-coordinate, then the maximum y-coordinate. Then are the x and y coordinates of the first point, and finally the x and y coordinates of the second point.

Output

For each pair of points, output the Manhattan distance, then the Winnipeg distance. Notice that these may be the same. Output both distances on one line as follows:

"Manhattan: XXX, Winnipeg: YYY"

where XXX and YYY are the Manhattan and Winnipeg distances.

Sample Input	Sample Output
3	Manhattan: 11, Winnipeg: 3
11 10 2 7 7 1	Manhattan: 98, Winnipeg: 2
100 100 99 1 99 99	Manhattan: 2, Winnipeg: 2
1000 1000 500 500 501 501	

Judging Data for Problem 3

30

11 10 2 7 7 1

100 100 99 1 99 99

1000 1000 500 500 501 501

440 70 122 39 407 13

10440 10070 5122 5039 5407 5013

126 861 68 709 71 664

10126 10861 5068 5709 5071 5664

135 415 14 359 109 77

785 614 701 602 517 517

83 259 74 217 66 168

973 998 527 352 800 422

387 739 312 31 141 370

478 495 393 176 284 386

51 546 3 425 18 401

198 401 77 261 126 388

457 78 50 57 88 24

560 369 319 280 18 244

541 174 198 27 427 149

94 331 10 48 11 19

800 287 197 35 667 117

786 44 684 40 506 26

638 243 18 205 443 229

559 475 257 67 32 305

833 721 682 350 713 627

377 517 298 46 217 22

210 326 119 53 85 212

112 974 79 569 29 159

839 570 793 380 476 530

434 425 312 341 235 107

310 229 86 60 184 144

Problem 4 – Kindergarten Policing

Arnold is starting his first day as a kindergarten teacher. Arnold is not really that great at his job, so he puts a number on the back of each student. This way, when Arnold gets all students to line up, he can figure out if any have wandered away. In a rare moment of brilliance, he also puts the highest number on the one student who never disappears anywhere, so he knows how many students he should have in line.

Fortunately, Arnold has figured out that if a child is missing, only one will be missing. Everyone may be there, but there will never be more than one child missing.

Unfortunately, the students rarely line up in order, so Arnold is having a hard time determining if any students are missing.

Write a program that reads in the student numbers and either prints out

"No one is missing."

or

"Student X is missing!"

where X is the number of the student not in line.

Input

The first line of the input gives the number of test cases to process. Each line after that has several numbers, separated by one space. The number of numbers in each line may differ (since some days, some children may be known to be absent). All numbers will be positive, starting with 1 for the first child in the class.

Output

Output one of the two messages for each line.

Sample Input	Sample Output
2	Student 4 is missing!!
1 2 3 5	No one is missing.
5 1 3 2 4	

Judging Data for Problem 4

7

1 2 3 5

5 1 3 2 4

5 4 3 2 1

1 2 3 4 5

10 1 2 3 7 8 5 6 4 9

10 1 2 3 7 8 5 6 4

6 5 3 1 2

Problem 5 – Stuck In Traffic

In this question, you will simulate the action of cars situated on a road. Each road will be given a sequence of empty spaces and cars (which are represented by their speed).

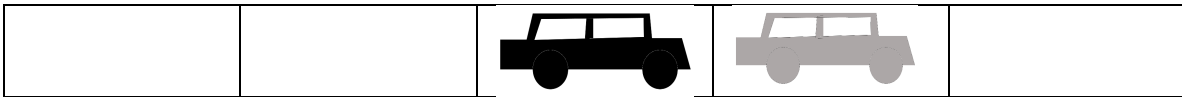
For instance, given a road that looks like this:

2 _ _ 0 _

the road has five "spaces", two of which are filled by cars:



Each car has a speed associated with it. If a car has speed v , then it will travel forward v spaces on the road. For instance, if the speed of the black car is 2 and the speed of the grey car is 0, then after one time step the road above would look like this:



In our text representation, that would be represented as

_ _ 2 0 _

Notice that cars travel from left to right in our model. **If a car reaches the end of the road, it is ignored. Car speeds will be set so that no car will have to pass another car during the update.**

Write a program that reads in one road configuration, moves the cars according to their speed, and outputs the next configuration of the road.

Input

The first line of the input will be a number that represents how many test cases your program has to process. Each line after that will represent one road. It will consist of underscore symbols `_` and numbers from 0 – 9, representing cars by their speeds. Note that there are no spaces on the lines (none between the numbers, between the underscore symbols or between numbers and underscores).

Output

For each road output the new configuration of cars, representing each car by their speed (which is the same as the speed before the movement) and blank spots on the road by the underscore symbol `_`. The road has the same length before and after the movement of cars.

Sample Input	Sample Output
3	<code>__20_</code>
<code>2__0_</code>	<code>__2__</code>
<code>2__2_</code>	<code>_11_0</code>
<code>11__0</code>	

Judging Data for Problem 5

29
000
001
011
020
300

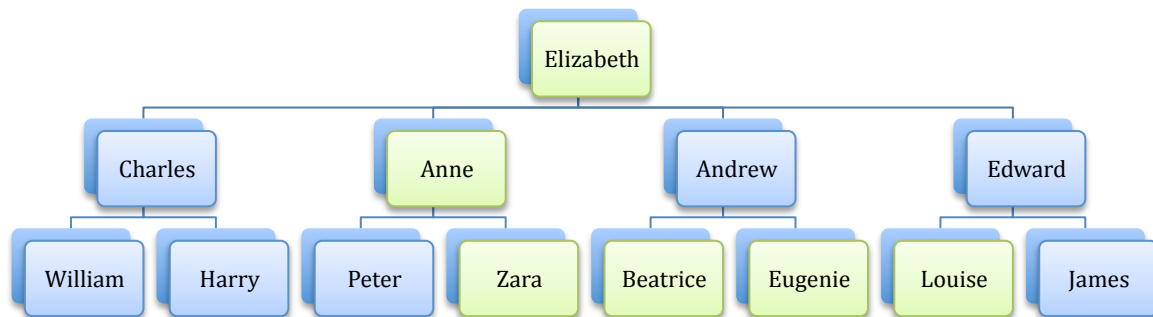
0__
1__
2__
3__
0
1
2
3
0_1
0_2__
0_3___

5__9____8____1.7____6____9____6____9____02__
5__9____8____1.7____6____9____6____9____02__
_2_8____7____8____9____7____6____2_4_6____3_4_2__
__1_3_01.9____0_1.6____4_6____4_1.1__
__1_3_01.9____0_1.6____4_6____4_5__
3_3_2_4____2_4____8____7____5_2_6____9____
2__2_9____01_3____4_4____1_9____5_2__
6____5____9____4_9____4_3_2_4____3_9____7____9__
9____4____7____9____4_1_0_06____8____9____1_6____4____
_1_4_5_5____2_05____2_3_6____4_5_9____5_7____9__
1_1_9____2_9____3_9____6____9____3_9____7____7__
2_5____8____2_7____0.3_9____1_3_2_0.9____7____

Problem 6 – Male-Preferred Primogeniture

For some royal families, who will be king or queen is determined by "male-preferred primogeniture", which means that male children will be king (regardless of age) before any female children will be queen, and that children of the queen or king will succeed to the throne before any of the younger brothers or sisters of that queen or king.

For instance, consider (a part of) the current British royal family, depicted below. We do not consider any spouses in this diagram, since we're only concerned with relationships between parents and children:



In this tree, children are ordered from left-to-right based on their age; Charles is the oldest child of Elizabeth while Edward is the youngest.

After Queen Elizabeth, the next in line to the throne are Charles (male), William (male), Harry (male), Andrew (male), Beatrice (female), Eugenie (female), Edward (male), James (male), Louise (female), Anne (female), Peter (male) and then Zara (female).

Write a program that takes a list of parent-child relationships and then prints out the line of succession (the order in which people will take the throne) for that family.

Input

There will be several test cases in each file. Each test case will start with a number on a line by itself, with 1 being the first case. Each name in the family will be unique, with no spaces. Each name is followed by the gender of the person (M for male and F for female).

For each family, the person currently king or queen is the first person listed on the first line.

For each person listed, if they have children, they are listed after their name, separated by a colon. For instance, Queen Elizabeth would be listed as

Elizabeth (F): Charles (M), Anne (F), Andrew (M), Edward (M)

The oldest child is listed first after the colon, followed by each child in order of his or her age.

A family member who is a parent will not be listed before their parent has already been listed, except for the current king or queen.

Output

For each case, first output "Case X" where X is the number of the case. Then print the names of each of the family members, one per line, in order of their succession to the throne. Do not print out their genders.

Sample Input	Sample Output
1 Elizabeth (F): Charles (M), Anne (F), Andrew (M), Edward (M) Charles (M): William (M), Harry (M) Anne (F): Peter (M), Zara (F) Peter (M): Savannah (F), Isla (F) Andrew (M): Beatrice (F), Eugenie (F) Edward (M): Louise (F), James (M)	Case 1 Elizabeth Charles William Harry Andrew Beatrice Eugenie Edward James Louise Anne Peter Savannah Isla Zara

Judging Data for Problem 6

1

JohnI (M) : JohnII (M)

2

JohnI (M) : JohnII (M), JaneI (F)

3

JohnI (M) : JaneI (F), JohnII (M)

4

JaneI (F): JaneII (F), JohnI (M)

JaneII (F): AliceI (F)

5

Elizabeth (F): Charles (M), Anne (F), Andrew (M), Edward (M)

Charles (M): William (M), Harry (M)

Anne (F): Peter (M), Zara (F)

Peter (M): Savannah (F), Isla (F)

Andrew (M): Beatrice (F), Eugenie (F)

Edward (M): Louise (F), James (M)

6

AerysII (M): Rhaegar (M), ViserysIII (M), Danerys (F)

Rhaegar (M): Rhaenys (F), Aegon (M)

7

Edwyle (M): Rickard (M)

Rickard (M): Brandon (M), Ned (M), Benjen (M), Lyanna (F)

Ned (M): Robb (M), Sansa (F), Arya (F), Brandon (M), Rickon (M)

(test cases continue on next page)

8

JaneI (F): JaneII (F), JohnI (M)
JaneII (F): AliceI (F)
JohnI (M): JaneIII (F), JohnII (M)
JaneIII (F): AliceII (F)
JohnII (M): JaneIV (F), JohnIII (M)
JaneIV (F): AliceIII (F)
JohnIII (M): JaneV (F), JohnIV (M)
JaneV (F): AliceIV (F)
JohnIV (M): JaneVI (F), JohnV (M)
JaneVI (F): AliceV (F)
JohnV (M): JaneVII (F), JohnVI (M)
JaneVII (F): AliceVI (F)
JohnVI (M): JaneVIII (F), JohnVII (M)
JaneVIII (F): AliceVII (F)
JohnVII (M): JaneIX (F), JohnVIII (M)
JaneIX (F): AliceVIII (F)
JohnVIII (M): JaneX (F), JohnIX (M)
JaneX (F): AliceIX (F)
JohnIX (M): JaneXI (F), JohnX (M)
JaneXI (F): AliceX (F)
JohnX (M): JaneXII (F), JohnXI (M)
JaneXII (F): AliceXI (F)
JohnXI (M): JaneXIII (F), JohnXII (M)
JaneXIII (F): AliceXII (F)
JohnXII (M): JaneXIV (F), JohnXIII (M)
JaneXIV (F): AliceXIII (F)

9

A (M): B (M), C (M), D (M)
B (M): E (M), F (M), G (F)
C (M): H (M), I (F), J (M)
D (M): K (M), L (F), M (F)
E (M): N (F), O (M), P (M)
F (M): Q (F), R (M), S (F)
G (F): T (F), U (F), V (M)
H (M): W (F), X (F), Y (F)
I (F): Ichild (M)
J (M): Jchild (M)
K (M): Kchild (M)
L (F): Lchild (M)
M (F): Mchild (M)
N (F): Nchild (M)
O (M): Ochild (M)
P (M): Pchild (M)
Q (F): Qchild (M)
R (M): Rchild (M)
S (F): Schild (M)
T (F): Tchild (M)
U (F): Uchild (M)
V (M): Vchild (M)
W (F): Wchild (M)
X (F): Xchild (M)
Y (F): Ychild (M)