LAB-PRAC-03_TYPES

FIFA Fever (p1v1d1)

FIFA Fever [10 marks]

Problem Statement
FIFA World Cup is a global footballing event in which nations compete to win the coveted title of World Champion. This yearâ€™s cup was held in Russia and was a delight to watch from a neutral point of view. Not so much if you were wishing for the biggies to win. There were so many surprises in this yearâ€™s edition: Germany getting knocked out at group stage, Argentina being humbled by eventual winners France, and Croatia showing grit and will to reach their maiden Final.

In the first line of input you are given an integer representing the team. You are then provided the scores of three matches of that team: a - b, c - d, e â€“ f where a, c, e represent goals scored by the team, and b, d, f represent goals scored against them. You need to compute the goal difference: total goals scored by them minus the total goals scored
against them.

Caution

1. Do not use any libraries other than stdio.h
2. Be careful of extra/missing spaces and extra/missing lines. Use the visible test cases to guide yourself.

INPUT:
X
a - b, c - d, e - f
OUTPUT:
Goal Difference for Team X: goaldifference

EXAMPLE:
INPUT

OUTPUT:

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 marks. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
7 1 - 1, 0 - 3, 2 - 1	Goal Difference for Team 7: -2
0 0 - 0, 0 - 0, 1 - 1	Goal Difference for Team 0: 0
1 7 - 1, 5 - 0, 3 - 3	Goal Difference for Team 1: 11
2 15 - 1, 3 - 0, 0 - 1	Goal Difference for Team 2: 16
625 1 - 1, 0 - 0, 0 - 1	Goal Difference for Team 625: -1
11257 12 - 11, 99 - 2, 1 - 22	Goal Difference for Team 11257: 77

Matrix Math (p1v2d1)

Matrix Math [20 marks]

Problem Statement
You are given two 3 x 3 floating point matrices A and B. The rows of the matrices will be given in different lines, one row in each line and there will be a blank line between the two matrices. The different entries of the matrix will be separated by a single space. Letâ€™s define matrices C and D as follows:
C = A + B
D = A x B
You have to print C and D as described in the output format. The elements of the output matrices should have only 2 digits beyond decimal point.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Print each row of the matrices C and D on **separate lines**, with a single space between 2 matrix entries in the same row.
3. Include **two blank lines** between printing the matrices C and D.

HINT: To print a floating point number, we use the notation %f in printf statements. To print a floating point number to only 2 digits beyond the decimal point, use the notation %0.2f in your printf statements.

INPUT:
A11 A12 A13
A21 A22 A23
A31 A32 A33

B11 B12 B13
B21 B22 B23
B31 B32 B33

OUTPUT:

C11 C12 C13
C21 C22 C23
C31 C32 C33

D11 D12 D13
D21 D22 D23
D31 D32 D33

EXAMPLE:
INPUT
0 0 0
0 0 0
0 0 0

1 1 1
1 1 1
1 1 1

OUTPUT:
1.00 1.00 1.00
1.00 1.00 1.00
1.00 1.00 1.00

0.00 0.00 0.00
0.00 0.00 0.00
0.00 0.00 0.00

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. Printing each line of the output (there will be total of 8 lines in your output) will carry equal weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible test cases and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output

0 0 0
0 0 0
0 0 0
 	
1 1 1
1 1 1
1 1 1
	1.00 1.00 1.00
	1.00 1.00 1.00
	1.00 1.00 1.00
	
	0.00 0.00 0.00
	0.00 0.00 0.00
	0.00 0.00 0.00

1 0 0
0 1 0
0 0 1
	2.00 0.00 0.00
	0.00 2.00 0.00
	0.00 0.00 2.00

1 0 0
0 1 0
0 0 1
	1.00 0.00 0.00
	0.00 1.00 0.00
	0.00 0.00 1.00

1 1 1
1 1 1
1 1 1
 	
1 1 1
1 1 1
1 1 1
	2.00 2.00 2.00
	2.00 2.00 2.00
	2.00 2.00 2.00
	
	3.00 3.00 3.00
	3.00 3.00 3.00
	3.00 3.00 3.00

1 2 3
4 5 6
7 8 9
 	
1 0 0
0 1 0
0 0 1
	2.00 2.00 3.00
	4.00 6.00 6.00
	7.00 8.00 10.00
	
	1.00 2.00 3.00
	4.00 5.00 6.00
	7.00 8.00 9.00

91 2.2 34.5
4.0 53.1 62.3
7.7 8.4 9.9
 	
11.1 13.3 12.4
32.3 99.9 91.3
12.1 22.2 89.9
	102.10 15.50 46.90
	36.30 153.00 153.60
	19.80 30.60 99.80
	
	1498.61 2195.98 4430.81
	2513.36 6740.95 10498.40
	476.58 1161.35 1752.41

9100.1 1.234 34.5
4.0 53.1 6002.3
765.47 812.434 9876.9
 	
11.71 13.23 102.4
32.13 909.9 91.31
102.1 22.12 89.95
	9111.81 14.46 136.90
	36.13 963.00 6093.61
	867.57 834.55 9966.85
	
	110124.27 122280.27 935066.12
	614587.75 181139.48 545165.06
	1043498.69 967838.00 1040994.62

The Tale of Three Lines (p1v3d1)

The Tale of Three Lines [20 marks]

Problem Statement
You are given the equation of three lines in three variables. The equations will only contain integer values. Consider the 3 x 3 matrix formed by this system of equations. Find the three minors of this matrix corresponding to the three elements in the first row of this matrix (in order) as well as the determinant of this matrix. Output these four numbers in four different lines.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.

2. First output the value of the minor M_11 then M_12 then M_13 followed by the determinant of this matrix. Give all values on different lines. Recall that M_ij denotes the minor of the matrix formed by removing the ith row and jth column of the matrix.

INPUT:
ax + by + c = 0
dx + ey + f = 0
gx + hy + i = 0

OUTPUT:
M11
M12
M13
Determinant

EXAMPLE:
INPUT
0x + 1y + 0 = 0
1x + 0y + 0 = 0
1x + 1y + 1 = 0

OUTPUT:
0
1
1
-1

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. Printing each line of the output (there will be total of4 lines in your output) will carry equal weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible test cases and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
0x + 1y + 0 = 0 1x + 0y + 0 = 0 1x + 1y + 1 = 0	0 1 1 -1
20x + 1y + 10 = 0 1x + -20y + 10 = 0 1x + 1y + 1 = 0	-30 -9 21 -381
1x + 2y + 3 = 0 2x + 3y + 1 = 0	5 1

3x + 1y + 2 = 0	-7 -18
0x + 0y + 0 = 0 0x + 0y + 0 = 0 0x + 0y + 0 = 0	0 0 0 0
12x + -27y + 3 = 0 31x + 41y + 34 = 0 66x + -97y + 52 = 0	5430 -632 -5713 30957
10x + 0y + 0 = 0 0x + 100y + 0 = 0 0x + 0y + 1000 = 0	100000 0 0 1000000

Fiery FIFA Fever (p2v1d1)

Fiery FIFA Fever [10 marks]

Problem Statement
A Round Robin tournament is one in which a contestant meets all other contestants in turn. If each contestant plays all other contestants twice, then this is called double round robin tournament. Football leagues around the world follow the double round robin format, in which teams play all other teams twice, once at their own stadium, designated as â€˜Homeâ€™ match and once at their opponentâ€™s stadium, called â€˜Awayâ€™ match.

You will be given the goal data of two of the teams in this double round robin tournament in the format described below.
TeamID1 a1 - b1, c1 - d1
TeamID2 a2 - b2, c2 - d2
where TeamID1 is the integer ID for team1. a1 and c1 represent goals scored by team1 in its two matches, and b1, d1 represent goals scored against team1 in those two matches. Similarly for team2 with ID TeamID2.

A points table is a table which shows the points scored by all the teams in the tournament. You have to output this table for the two teams for whom you have been given data. In the following, GF stands for Goals For, GA stands for Goals Against, and GD stands for Goal Difference. Your table will have four columns, and three rows:

1. The first row will contain the text headings ID, GF, GA, and GD.
2. The second row with contain the text ID of team1, followed by the total goals scored by team1, followed by total goals scored against team1, followed by the goal difference.
3. The third row with the same details, but for team2.
4. You have to ensure that the first characters in each of the columns should be aligned vertically with the corresponding column heading. Use the tab character (with escape sequence â€˜\tâ€™) between ID and GF, GF and GA and GD columns to achieve this.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Press Evaluate/Submit once without writing any code to see how the output is expected.
3. Use the tab character instead of space between two entries of two side-by-side columns.

INPUT:
ID_1 a - b, c - d
ID_2 w - x, y - z

OUTPUT:
ID GF GA GD
ID_1 (a + c) (b + d) (gd1)
ID_2 (w + y) (x + z) (gd2)

EXAMPLE:
INPUT
7 1 - 1, 0 - 0
5 0 - 0, 2 - 1

OUTPUT:
ID GF GA GD
7 1 1 0
5 2 1 1

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 marks. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
7 1 - 1, 0 - 0 5 0 - 0, 2 - 1	ID GF GA GD 7 1 1 0 5 2 1 1
7 25 - 1, 0 - 6 5 16 - 5, 3 - 1	ID GF GA GD 7 25 7 18 5 19 6 13
17 5 - 1, 0 - 6 51 6 - 5, 3 - 1	ID GF GA GD 17 5 7 -2 51 9 6 3
82 11 - 1, 232 - 325 23 223 - 23, 5 - 5	ID GF GA GD 82 243 326 -83 23 228 28 200
100 1 - 1, 2 - 3 101 15 - 1, 12 - 0	ID GF GA GD 100 3 4 -1 101 27 1 26
300 106 - 11, 200 - 300 401 15 - 71, 12 - 90	ID GF GA GD 300 306 311 -5

	401 27 161 -134

The Final Rational (p2v2d1)

The Final Rational [20 marks]

Problem Statement
You are given a rational number with 6 digits after the decimal point. You have to output the first three and the last three digits of the number after the decimal place. For example if the number is 123.456789, then the output should be 456 and 789. Both outputs should be given on two separate lines.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Give both numbers on different lines
3. Note that you have to output the digits themselves i.e. if the last three digits are 0, 5 and 6, you have to output 056 and not 56.
4. Be careful that there is no guarantee how many digits the number has *before* the decimal point. Only the number of digits after the decimal point are known to be 6.

INPUT:
N
where N is a floating point value with 6 digits after the decimal.

OUTPUT:
First three digits after the decimal
Last three digits after the decimal

EXAMPLE:
INPUT
123.456789

OUTPUT:
456
789

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Giving correct output in each line carries 50% of the marks. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases. Ensure that there are no missing/extra spaces or missing/extra lines in your output.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
9.100000	100 000
123.456789	456 789
0.123123	123 123
0.000001	000 001
1000.989898	989 898
000.000000	000 000

Quadratic Quandary (p2v3d1)

Quadratic Quandary [20 marks]

Problem Statement
You will be given a **quadratic equation with integer coefficients** which is guaranteed to have real roots (maybe repeated) in the following form.
ax^2 + bx + c = 0
You need to compute the following four quantities with respect to the given quadratic equation

1. The discriminant of the quadratic equation
2. The point where the quadratic curve described by the equation attains its maximum or minimum i.e. if a > 0, find the point where the curve attains its minimum and if a < 0, find the point where the curve attains its maximum.
3. Sum of the two roots of the equation
4. Absolute difference of the two roots of the equation

Print each result on a different line. Print each result **correct to three decimal places**.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Give all four numbers on different lines.
3. Print all results correct only to three decimal places.
4. The quadratic equation we give you will always have real roots but may have repeated roots
5. The equation will have integer coefficients but the roots, determinant etc may be non-integer.

Hint: The math.h library has been included for you. Use the fabs(x) function to calculate the absolute value of a float/double variable x. Use sqrt(x) to calculate the square root of a float or double variable x. Use the notation %0.3f or the shorthand %.3f to print a float/double variable correct to three decimal places.

INPUT:
ax^2 + bx + c = 0
where a, b, c are integers and a is non-zero.

OUTPUT:
Discriminant: D
Extremum: E
Sum: R1 + R2
Difference: |R1-R2|

EXAMPLE:
INPUT
1x^2 + 2x + -6 = 0

OUTPUT:
Discriminant: 28.00
Extremum: -1.00
Sum: -2.00
Difference: 5.29

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are four lines in your output. Giving correct output in each line carries 25% of the marks. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases. Ensure that there are no missing/extra spaces or missing/extra lines in your output.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
1x^2 + 2x + -6 = 0	Discriminant: 28.000 Extremum: -1.000 Sum: -2.000 Difference: 5.292
1x^2 + 5x + 6 = 0	Discriminant: 1.000 Extremum: -2.500 Sum: -5.000 Difference: 1.000
15311x^2 + -3234x + -10151 = 0	Discriminant: 632146624.000 Extremum: 0.106 Sum: 0.211 Difference: 1.642
3x^2 + 1x + -1 = 0	Discriminant: 13.000 Extremum: -0.167 Sum: -0.333 Difference: 1.202
14x^2 + 1x + 0 = 0	Discriminant: 1.000

	Extremum: -0.036 Sum: -0.071 Difference: 0.071
2x^2 + 4x + 2 = 0	Discriminant: 0.000 Extremum: -1.000 Sum: -2.000 Difference: 0.000

FIFA Fractions (p3v1d1)

FIFA Fractions [10 marks]

Problem Statement
The FIFA World Cup is a global footballing event in which nations compete to win the coveted title of World Champion. This yearâ€™s cup was held in Russia and was a delight to watch from a neutral point of view. Not so much if you were wishing for the biggies to win. There were so many surprises in this yearâ€™s edition: Germany getting knocked out at group stage, Argentina being humbled by eventual winners France, and Croatia showing grit and will to reach their maiden Final.

In the first line of input you are given the team number which will be an **integer** and in the second line, you are provided the scores of three matches played by that team: a - b, c - d, e - f where a, c, e are **integers** representing goals scored by that team, and b, d, f are **integers** representing goals scored against that team by the opponent team.

The goal ratio of a match is defined as the ratio of goal scored by the team to the goals scored against that team. You need to compute the **sum of goal ratios** of all three matches in the format given below. The output should be rounded off to 3 places beyond the decimal point.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. We are asking for the sum of the goal ratios of the three matches.

INPUT:
Team Number: x
a - b, c - d, e - f

OUTPUT:
Goal Ratio sum for Team x: ratio

EXAMPLE:
INPUT
Team Number: 10
1 - 1, 0 - 3, 2 - 1

OUTPUT:
Goal Ratio sum for Team 10: 3.000

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 marks. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
Team Number: 10 1 - 1, 0 - 3, 2 - 1	Goal Ratio sum for Team 10: 3.000
Team Number: 25 0 - 1, 2 - 4, 2 - 1	Goal Ratio sum for Team 25: 2.500
Team Number: 40 3 - 5, 4 - 5, 7 - 9	Goal Ratio sum for Team 40: 2.178
Team Number: 40 9 - 11, 13 - 15, 7 - 17	Goal Ratio sum for Team 40: 2.097
Team Number: 40 4 - 6, 3 - 8, 0 - 1	Goal Ratio sum for Team 40: 1.042
Team Number: 40 1 - 2, 3 - 6, 0 - 1	Goal Ratio sum for Team 40: 1.000

Digit Dilemma (p3v2d1)

Digit Dilemma [20 marks]

Problem Statement
You are given a 4-digit **integer number** N and another single digit **integer** A. On the first line, output the sum of digits of the number N. On the second line, output how many times does the digit A appear in the number N.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful that the number N may contain zeros (even at the beginning), as well as the digit A may itself be zero. For example if N = 0501, A = 0, then we will say that A appears **twice** in N
3. Give your outputs on two different lines

INPUT:
N A
OUTPUT:
sum of digits of N
number of times digit A appears in N

EXAMPLE:

```
INPUT
1223 2

OUTPUT:
8
2
```

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. Your output has two lines. Printing each line correctly will get you 50% marks. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
1223 2	8 2
1011 9	3 0
0000 0	0 4
1111 1	4 4
0890 0	17 2
0001 0	1 3

Recursive Recharge (p3v3d1)

Recursive Recharge [20 marks]

Problem Statement
Just as the factorial function is defined in terms of itself, we are defining a new function below. The function is defined on non-negative integers i.e. 0, 1, 2, 3, ...
$$f(0) = P$$$$f(n) = f(n-1)^2 + \frac{n\cdot m}{\sqrt v + e}$$
The values of m, v, e, P will be provided to you as **integers**. Your job is to output f(1), f(2), f(3), f(4) in **different lines**. All your outputs should be rounded off to to **three decimal places**

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. The inputs will be integers but the outputs may be non-integers.
3. Try using double values/double typecasting in case you are getting errors with float variables and float typecasting. Be careful that while %f works while printing doubles, when using scanf with doubles, always use %lf.
4. The integer v will always be given as positive but other integers M, e, P, may be negative.

HINTS:

1. The math.h library has been included for you as you may require it for the square root and other functions.

INPUT:
m, v, e, P

OUTPUT:
f(1)
f(2)
f(3)
f(4)

EXAMPLE:
INPUT
1, 2, 3, 2

OUTPUT:
4.227
18.317
336.182
113019.391

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are 4 lines in your output. Each will receive 25% of the total points if entered correctly. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
1, 2, 3, 2	4.227 18.317 336.182 113019.391
1, 1, 1, 1	1.500 3.250

	12.062 147.504
0, 1, 0, 0	0.000 0.000 0.000 0.000
-10, 2, 1, -2	-0.142 -8.264 55.868 3104.712
5, 2, 1, -1	3.071 13.574 190.456 36281.633
1, 4, -3, 1	0.000 -2.000 1.000 -3.000

Breaking the Lego Safe (p4v1d1)

Breaking the Lego Safe [10 marks]

Problem Statement
In last weekâ€™s lab we talked about a Lego safe in which your friend wanted to give you a secret digit. The safe is to be built using lego blocks. Last time you sent an image of the number in the safe to your friend as a proof of your trustworthiness. Seeing the image that you had sent, your friend agrees that you are indeed trustworthy, and now wants to use this safe for his task. He will provide two safes, each containing **an integer**, and you have to return the arithmetic mean of the given two numbers in a new safe. The new safe is constructed by replacing â€˜#â€™ with a character with a 'O' symbol (capital letter O). Note that for greater security, the bars of the new safe also have four more â€˜-â€™ symbols. The arithmetic mean stored in the safe should have **only one digit beyond the decimal point**.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. The arithmetic mean stored in the safe should have only one digit beyond the decimal point.

INPUT:
```
#-#-# #-#-#
#-a-# #-b-#
#-#-# #-#-#
```

OUTPUT:
```
O---O---O
O--C--O
O---O---O
```

EXAMPLE:

```
INPUT
#-#-# #-#-#
#-3-# #-4-#
#-#-# #-#-#

OUTPUT:
O---O---O
O--3.5--O
O---O---O
---------------------------------------------------------------------
```

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 marks. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
#-#-# #-#-# #-3-# #-4-# #-#-# #-#-#	O---O---O O--3.5--O O---O---O
#-#-# #-#-# #-10-# #-10-# #-#-# #-#-#	O---O---O O--10.0--O O---O---O
#-#-# #-#-# #-1-# #-2-# #-#-# #-#-#	O---O---O O--1.5--O O---O---O
#-#-# #-#-# #-30-# #-459-# #-#-# #-#-#	O---O---O O--244.5--O O---O---O
#-#-# #-#-# #-333-# #-777-# #-#-# #-#-#	O---O---O O--555.0--O O---O---O
#-#-# #-#-# #-0-# #-0-# #-#-# #-#-#	O---O---O O--0.0--O O---O---O

The Final Rational Revisited (p4v2d1)

The Final Rational [20 marks]

```
---------------------------------------------------------------------
```

Problem Statement
You are given a rational number N with 6 digits after the decimal point. You have to first output the number formed by the digits after the decimal place - call this number P. Then find the number formed by reversing the six digits (call this new number Q) and output the sum P+Q on **a different line**. For example if the

number is 123.456789, then P = 456789 and Q = 987654 and so the output should be 456789 and 1444443. Both outputs should be given on two separate lines.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Give both numbers on different lines
3. Note that you don't have to output the digits themselves i.e. if the digits after the decimal place are 0, 0, 5, 6, 1, 2, you have to output 5612 and not 005612.
4. Be careful that there is no guarantee how many digits the number has *before* the decimal point. Only the number of digits after the decimal point are known to be 6.

HINTS:

1. You may require the use of math.h functions for which math.h has been included for you. However, it is **not necessary to use math.h functions** to solve this problem.
2. If you are facing precision issues, **use a double variable** instead of float variable to perform calculations.

INPUT:
N
where N is a floating point value with 6 digits after the decimal

OUTPUT:
Number P formed out of digits after the decimal point
P + Q where Q is the number formed by reversing **all the digits** after the decimal point

EXAMPLE:
INPUT
123.456789

OUTPUT:
456789
1444443

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Giving correct output in each line carries 50% of the marks. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases. Ensure that there are no missing/extra spaces or missing/extra lines in your output.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
123.456789	456789 1444443
9.100000	100000 100001
0.456456	456456 1111110
0.000002	2 200002
10.989898	989898 1888887
000.000000	0 0

Developing an interest in interest (p4v3d1)

Developing an interest in interest [20 marks]

Problem Statement
Loans and interest are crucial to modern economy. The argument for interest is as follows: It is preferable to receive a given good now rather than in the future and therefore interest is compensation for the time the lender forgoes the benefit of spending the money. You are given the following quantities

1. principal amount P
2. annual rate of interest R (in %)
3. first duration of time in years T1
4. second duration of time in years T2
5. number of times interest is compounded per year N

All the above quantities will be provided to you as **integers**. You need to give the following outputs **on different lines**

1. total interest accrued if simple interest is applied for T1 years
2. if after T1 years of simple interest, we start applying compound interest for T2 years (compounded N times a year), then total compound interest accrued in those T2 years.
3. total interest accrued if we had just applied simple interest for T1 + T2 years
4. total interest accrued if we had just applied compound interest for T1 + T2 years (compounded N times a year)

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. All outputs ask for interest amounts, not final amounts.
3. All the **outputs should be rounded to 3 places** beyond the decimal point.
4. Use double variables for all your computations. **Do not use float** as it may cause you loss of precision.

HINT: You may require the use of functions from the math.h library which has been included for you. The formula for calculating the **final amount** if interest is compounded N times a year for T years at R% rate with principal P is $$P*(1+R/(100*N))^{N*T}$$

EXAMPLE:

```
INPUT
P = 1000
R = 10
T1 = 1
T2 = 1
N = 1

OUTPUT:
100.000
110.000
200.000
210.000
```

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There will be 4 lines in your output. Each carries 25% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output

P = 1000
R = 10
T1 = 1
T2 = 1
N = 1
	100.000
	110.000
	200.000
	210.000

P = 1000
R = 10
T1 = 0
T2 = 3
N = 1
	0.000
	331.000
	300.000
	331.000

P = 1000
R = 7
T1 = 3
T2 = 3
N = 1
	210.000
	272.302
	420.000
	500.730

P = 1000
R = 7
T1 = 3
T2 = 3
N = 4
	210.000
	280.042
	420.000
	516.443

P = 123456
R = 9
T1 = 3
T2 = 10
N = 12
	33333.120
	227556.999
	144443.520
	272585.551

P = 999
R = 100
T1 = 5
T2 = 5
N = 2
	4995.000
	339650.244
	9990.000
	3320932.473