LAB-PRAC-04_COND

Trouble with Triangles (p1v1d1)

Trouble with Triangles [10 marks]

Problem Statement
You will be provided the side lengths of a triangle in increasing order, **separated by commas**. Side lengths will be **strictly positive integers**. Depending on the input, you have to output **only one** of the following labels

1. **Invalid**: if the lengths provided cannot belong to a triangle. Remember that 3 positive integers can form the sides of a valid triangle if and only if the sum of any two of the side lengths is larger than or equal to the length of the third side. That means that if we have side lengths a, b, c, then for these to form a triangle, we must have
   $$\begin{align}a + b &\geq c\\b + c &\geq a\\c + a &\geq b\end{align}$$
   The above inequalities are often called the **triangle inequalities**.
2. **Equilateral**: if all sides of the triangle are equal
3. **Isosceles**: if any two sides of the triangle are equal
4. **Scalene**: if all sides of the triangle are unequal

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Output only one label. If you feel that more than one label applies to the triangle, please follow the rules given below.
    1. If the side lengths are invalid, for example 3, 4, 8, print only "Invalid" (without quotes) for this case. Do not print "Scalene" or "Isosceles" for such examples since they do not form valid triangles.
    2. If all side lengths are equal, for example 5, 5, 5, print only "Equilateral" in this case. Do not print "Isosceles" in this case.
3. Side lengths will be integers and be given in increasing order.

INPUT:
length1, length2, length3

OUTPUT:
Label
EXAMPLE:
INPUT
3, 4, 5

OUTPUT:
Scalene

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, 5	Scalene
3, 4, 9	Invalid
3, 3, 9	Invalid
20, 20, 20	Equilateral
20, 40, 60	Scalene
2, 9, 9	Isosceles

Ms- Mathematica (p1v2d1)

Ms. Mathematica [20 marks]

Problem Statement
Ms. Mathematica really loves numbers but some numbers are more loved than others. Currently, she likes complex numbers which have special real and imaginary coefficients. She needs your help to find out if a given complex number is her favorite or not.

You will be given a complex number n = a + bi in a format described below, with both the real and imaginary parts being **4-digit integers** and an integer m, on **two separate lines**. Your job is to output the following in **two different lines**

1. The sum of the digits of the real part a of the number n and the product of the digits of the imaginary part b of the number n. Both numbers should be output on the same line, **separated by a comma and a space**
2. If both the sum and product values you calculated above are divisible by m then print â€œFavorite" (without the quotes), else print "Not favorite" (without the quotes).

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Both the sum and the product should be output on the same line (the first line) and be separated by a single comma and a single space.
3. Output the sum and product as integers with no leading zeros or decimal points etc.
4. The four digit numbers we give you as a and b, may have zeros in their representation.

INPUT:
a + bi
m

OUTPUT:
sum, product
message

EXAMPLE:
INPUT
1234 + 5678i
5

OUTPUT:
10, 1680
Favorite

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

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% 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
1234 + 5678i 5	10, 1680 Favorite
9455 + 3498i 23	23, 864 Not favorite
0002 + 1112i 2	2, 2 Favorite
7322 + 5764i 7	14, 840 Favorite
4563 + 5432i 25	18, 120 Not favorite
1234 + 5678i 4	10, 1680 Not favorite

Pollution Problem (p1v3d1)

Pollution Problem [20 marks]

Problem Statement
The Kanpur city has a growing problem of particulate matter (PM) pollution. IITK researchers are trying to help measure and correct the problem. You will be given four quantities in **four different lines** in a format described below -- all four quantities will be **non-integer**. On the first line, we will give you a threshold, and then on the next three lines we will give you PM pollution levels in Kanpur for Monday, Tuesday and Wednesday, say PM1, PM2 and PM3. **Assume that the PM pollution level on Sunday is always 1.0** (since being a holiday PM levels are less).

You have to output the following on **four different lines**.

1. For each day Mon, Tue, Wed, if the PM level on that day minus the level on the previous day is **greater than or equal to** the threshold, print High (see below for exact format) else print Low.
2. Print the output for every day on a different line.
3. For Monday, take Sunday as the previous day for which PM level is given above.
4. If the number of days where your answer above was high is **less than** two, output 0.000 on the fourth line. If the number of such days is **greater than or equal to** two, then print the following quantity **rounded to three decimal places** in the fourth line
$$\sqrt{\frac{PM1^2 + PM2^2 + PM3^2}3}$$

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Using float variables to read and perform calculations should be sufficient. In other words, you should not require the use of double variables in this question.
3. Although you may use copy (CTRL+C) and paste (CTRL+V) to avoid typing similar code again and again, be careful to not make mistakes while doing so.

INPUT:
threshold
Monday: PM1
Tuesday: PM2
Wednesday: PM3

OUTPUT:
Monday: message
Tuesday: message
Wednesday: message

EXAMPLE:
INPUT
0.1
Monday: 1.2
Tuesday: 1.4
Wednesday: 1.6

OUTPUT:
Monday: High
Tuesday: High

Wednesday: High
1.409

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

There will be partial grading in this question. There are four lines in your output. Printing each line correctly, in the correct order, 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
0.1 Monday: 1.2 Tuesday: 1.4 Wednesday: 1.6	Monday: High Tuesday: High Wednesday: High 1.409
0.5 Monday: 0.0 Tuesday: 0.0 Wednesday: 1.0	Monday: Low Tuesday: Low Wednesday: High 0.000
1.2 Monday: 14.56 Tuesday: 11.22 Wednesday: 15.78	Monday: High Tuesday: Low Wednesday: High 13.987
0.0 Monday: 1.0 Tuesday: 1.0 Wednesday: 1.0	Monday: High Tuesday: High Wednesday: High 1.000
1.2 Monday: 1.2 Tuesday: 1.3 Wednesday: 2.49	Monday: Low Tuesday: Low Wednesday: Low 0.000
0.0 Monday: 0.0 Tuesday: 0.0 Wednesday: 0.0	Monday: Low Tuesday: High Wednesday: High 0.000

In or Out (p2v1d1)

In or Out [10 marks]

Problem Statement
You will be given the 2D coordinates of the center of a circle and its radius in a format given below. You will also be given the 2D coordinates of another point. You have to print "In" (without quotes) if the point is

inside the boundary of the circle, "On" (without quotes) if the point is on the boundary of the circle, and "Out" (without quotes) if the point is outside the boundary of the circle. All coordinates and radii will be **integers**.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful that although the coordinates can be stored as int variables, your calculations may involve values that are too large for the int variable. Take care and use appropriate datatypes.

INPUT:
(xcenter, ycenter), r = radius
(x, y)

OUTPUT:
Label (In/On/Out)
EXAMPLE:
INPUT
(7, 6), r = 4
(3, 3)

OUTPUT:
Out

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 points. 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
(10000, 10000), r = 50000 (40000, 50000)	On
(7, 6), r = 4 (3, 3)	Out
(111, 111), r = 121 (41, 23)	In
(23456789, 23456789), r = 1000 (23456789, 23456789)	In
(1, 1), r = 5 (4, 5)	On
(1000000, 1000000), r = 100 (2, 2)	Out

Rick-s Number (p2v2d1)

Rick's Number [20 marks]

Problem Statement
You will be given two **non-negative four digit integers**, call them m = abcd and n = pqrs. Given these two numbers, we define the Rick's number as follows

1. If the sum of the digits of **exactly one of m and n** are even, then the Rick's number is defined to be the number with the digits apbqcrds.
2. Otherwise, the Rick's number is defined to be the number with the digits paqbrcsd.

There will be **four lines** in your output

1. In the first line of your output, you have to print the sum of the digits of m
2. In the second line of your output, you have to print the sum of the digits of n
3. In the third line of your output, you have to print "Both" (without quotes) if both the above sums are even, "One" (without quotes) if exactly one of the sums is even, and "Neither" (without quotes) if neither of the sums is even.
4. In the fourth line of your output, you have to print the Rick's number **as an integer** without any leading zeros i.e. if the Rick's number is 02145698, then print 2145698.

Caution

1. The digits of the number m and n may contain zeros
2. The Rick's number has to be output as an integer without any leading zeros. As mentioned above, if the Rick's number is 02145698, then print 2145698.
3. Be careful about extra/missing lines and extra/missing spaces.

INPUT:
m n

OUTPUT:
sum1
sum2
Both/One/Neither
Rick'sNumber

EXAMPLE:
INPUT
1234 5678

OUTPUT:
10
26
Both
51627384

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

There will be partial grading in this question. There are four lines in your output. The first two lines carry 12.5% weightage, the third line carries 25% weightage and the last line carries 50% weightage. Printing each line correctly, in the correct order, will give you the corresponding partial 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
1234 5678	10 26 Both 51627384
0234 1638	9 18 One 1263348
5433 5877	15 27 Neither 55847373
0000 1111	0 4 Both 10101010
3333 0000	12 0 Both 3030303
5079 0103	21 4 One 50017093

Its Tax Time (p2v3d1)

Its Tax Time [20 marks]

Problem Statement
In a week's time, all Indian citizens who are income tax assessees will have to file their tax returns (your instructor is one of them). Can you help your instructor file his taxes? The Indian tax code for personal

income is a bit more complicated so we will use a simplified tax code below. Let the income of Mr C be Rupees X

1. Slab 1: If X is less than or equal to 100,000, no tax is to be paid
2. Slab 2: If X is greater than 100,000 but less than or equal to 200,000, then tax paid is 10% of (X-100,000)
3. Slab 3: If X is greater than 200,000 but less than or equal to 300,000, then tax paid is 10,000 + 20% of (X-200,000)
4. Slab 4: If X is greater than 300,000, then tax paid is 30,000 + 30% of (X-300,000)

You will be given an income value as a **positive integer**. You have to give output in **two lines**. In the first line you have to output which Slab does the income fall in and in the second line you have to output the total tax **rounded off to two decimal places**.

Caution

1. The commas in the amounts above are just for ease of reading. Do not put commas in numbers when coding. Mr. C does not understand commas inside integers.
2. Be careful about extra/missing lines and extra/missing spaces.
3. Carefully read the tax code. Do not make mistakes between "greater than" and "greater than or equal to"

INPUT:
income

OUTPUT:
Slab
Tax

EXAMPLE:
INPUT
250000

OUTPUT:
Slab 3
20000.00

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

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% 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
250000	Slab 3 20000.00
123456789	Slab 4 36977036.70
100000	Slab 1 0.00
200001	Slab 3 10000.20
300003	Slab 4 30000.90
199999	Slab 2 9999.90

The Toppers (p3v1d1)

The Toppers [10 marks]

Problem Statement
You will be given the roll number and marks (out of 100) for three students in a format described below. The roll numbers **will not repeat** as well as **no two students will have the same marks**. All roll numbers and marks will be given **as positive integers**. You have to print the roll number of the students with the highest and the second highest marks. Both roll numbers should be on the same line, and be separated by a comma and a single space.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful that you have to output the roll numbers and not the marks.

INPUT:
(roll1, marks1) (roll2, marks2) (roll3, marks3)

OUTPUT:
rollHighest, rollNextHighest

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 points. 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
(1011, 25) (1211, 47) (1098, 99)	1098, 1211
(1011, 100) (1211, 60) (1098, 40)	1011, 1211
(12011, 21) (12211, 55) (12098, 15)	12211, 12011
(150151, 43) (152151, 87) (150958, 51)	152151, 150958
(16151, 77) (16251, 12) (16958, 45)	16151, 16958
(11153, 77) (11253, 66) (11953, 99)	11953, 11153

Isotonic Regression (p3v2d1)

Isotonic Regression [20 marks]

Problem Statement
You will be given 5 (x,y) coordinate pairs (**all coordinates will be integers**), say (x1, y1) till (x5, y5) in a format described below. The values x1, x2, x3, x4, x5 will always be unique and given in increasing order. Given this, we define the following function

$$
\begin{align}
f(x1) &= y1\\
f(x2) &= \begin{cases}
y2 &\mbox{if } y2 \geq f(x1) \\
f(x1) &\mbox{if } y2 < f(x1)\end{cases}\\
f(x3) &= \begin{cases}
y3 &\mbox{if } y3 \geq f(x2) \\
f(x2) &\mbox{if } y3 < f(x2)\end{cases}\\
f(x4) &= \begin{cases}
y4 &\mbox{if } y4 \geq f(x3) \\
f(x3) &\mbox{if } y4 < f(x3)\end{cases}\\
f(x5) &= \begin{cases}
y5 &\mbox{if } y5 \geq f(x4) \\
f(x4) &\mbox{if } y5 < f(x4)\end{cases}
\end{align}
$$

Your job is to output, on **four different lines** the values f(x2), f(x3), f(x4) and f(x5) in the format described below. All values of f will be **integers**.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful that you do not have to print f(x1). You only have to print f(x2), f(x3), f(x4) and f(x5), and that too on 4 different lines using the format described below.
3. Although x1, x2 etc are unique, y1, y2 etc may not be unique.

INPUT:
(x1, y1) (x2, y2) (x3, y3) (x4, y4) (x5, y5)

OUTPUT:
f(value of x2) = f(x2)
f(value of x3) = f(x3)
f(value of x4) = f(x4)
f(value of x5) = f(x5)

EXAMPLE:
INPUT
(1, 1) (2, 4) (3, 9) (4, 4) (5, 1)

OUTPUT:
f(2) = 4
f(3) = 9
f(4) = 9
f(5) = 9

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

There will be partial grading in this question. There are four lines in your output. Printing each line correctly, in the correct order, 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
(1, 1) (2, 4) (3, 9) (4, 4) (5, 1)	f(2) = 4 f(3) = 9 f(4) = 9 f(5) = 9
(1, 1) (2, 4) (3, 9) (4, 16) (5, 25)	f(2) = 4 f(3) = 9 f(4) = 16 f(5) = 25
(11, 1) (21, 4) (31, 4) (41, 2) (51, 44)	f(21) = 4 f(31) = 4 f(41) = 4 f(51) = 44
(101, 66) (201, 55) (301, 44) (401, 33) (501, 22)	f(201) = 66 f(301) = 66 f(401) = 66 f(501) = 66
(110, 66) (210, 66) (310, 440) (410, 33) (510, 2200)	f(210) = 66 f(310) = 440

	f(410) = 440 f(510) = 2200
(110, 66) (2100, 77) (3010, 88) (4010, 99) (5100, 22)	f(2100) = 77 f(3010) = 88 f(4010) = 99 f(5100) = 99

Super Leap Years (p3v3d1)

Super Leap Years [20 marks]

Problem Statement
Suppose we have 3 positive numbers A, B, C, and are given a year Y. Using these we define 3 rules as given below

1. **Rule 1**: If Y is divisible by A, we **call** Y a super leap year
2. **Rule 2**: If Y is divisible by A but is also divisible by B, we **don't call** Y a super leap year
3. **Rule 3**: If Y is divisible by A but is also divisible by both B and C, we **call** Y a super leap year

Any year that is not called a super leap year is called a super common year.

You will be given a year Y, as a **positive integer**, as well as four numbers, A, B, C, N, all four given as **positive integers**, in a format given below. **The number N will be either 1, 2 or 3**. You have to output the following two labels in two different lines.

1. Apply rules 1, 2, and 3 with Y, A, B, C to find if Y is called a super leap year or not. If it is a super leap year, print "Leap" (without quotes) else print "Common" (without quotes).
2. Apply only rules from rule 1 till rule N with Y, A, B, C to find if Y is called a super leap year or not. If it is a super leap year, print "Leap" (without quotes) else print "Common" (without quotes). Remember that the number N will be either 1, 2 or 3.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. For example, if N = 2, then in the second line, you only have to apply rules 1 and 2 to find out if Y is a super leap year or not. If N = 1, then in the second line, you only have to apply rule 1 to find out if Y is a super leap year or not.
3. In the first line, all three rules must be applied to find out whether the year is a super leap year or not, irrespective of the value of N.
4. Note that there are no restrictions on A, B, C other than that they are positive integers. A, B, C need not be in increasing or decreasing order. A need not divide B or C or vice versa etc etc.

INPUT:
Y A B C N
OUTPUT:

Leap/Common
Leap/Common

EXAMPLE:
INPUT
2000 4 100 400 2

OUTPUT:
Leap
Common

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

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% 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
2000 4 100 400 2	Leap Common
55 7 13 5 1	Common Common
40 7 4 5 2	Common Common
1800 4 100 400 1	Common Leap
1280 4 8 64 2	Leap Common
12 4 8 64 2	Leap Leap

Make Room for Rectangles (p4v1d1)

Make Room for Rectangles [10 marks]

Problem Statement
You will be given the coordinates of the bottom-left and top-right corners of an axis-aligned rectangle (i.e. a rectangle whose sides are parallel to the x and y axes). The input format is given below. The 4 quadrants on the plane are defined below for your convenience

1. Quadrant I: x >= 0, y >= 0

2. Quadrant II: x < 0, y >= 0
3. Quadrant III: x < 0, y < 0
4. Quadrant IV: x >= 0, y < 0

You will need out output the number of quadrants with which this rectangle intersects. The coordinates will be given as **integer** numbers.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful about using relational operators.

INPUT:
(xbotleft, ybotleft) (xtopright, ytopright)

OUTPUT:
number

EXAMPLE:
INPUT
(-1, -1) (2, 2)

OUTPUT:
4

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 points. 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
(1, 1) (5, 5)	1
(-1, -1) (2, 2)	4
(-6, -6) (1, -2)	2
(0, 0) (1, 2)	1
(-1, 0) (0, 1)	2
(-2, -2) (0, 0)	4

Quadratic Quandry Revisited (p4v2d1)

Quadratic Quandry Revisited [20 marks]

Problem Statement
You are given a quadratic equation with **integer** coefficients in a format described below. You have to find out and output various quantities about this quadratic equation as given below.

1. In the first line of your output, print "Real" (without quotes) if the equation has real roots else print "Complex" (without quotes) if the equation has complex roots.
2. In the second line of your output, print the discriminant of the equation (it will also be an integer) $$D = b^2 - 4ac$$
3. In the third line of your output, print the root which you get as $$\frac{-b + \sqrt{D}}{2a}$$
4. In the fourth line of your output, print the root which you get as $$\frac{-b - \sqrt{D}}{2a}$$

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. If a root is real, output it **rounded off to 3 decimal places**.
3. If the root is imaginary, output it in the following format
   r + qi or r - qi
   where r and q should be **rounded off to 3 decimal places**
4. If the equation has repeated roots, you have to output the same root twice in both the third and the fourth lines.

INPUT:
ax^2 + bx + c

OUTPUT:
Real/Complex
Discriminant
Root1
Root2

EXAMPLE:
INPUT
1x^2 + 4x + 3

OUTPUT:
Real
4
-1.000
-3.000

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

There will be partial grading in this question. There are four lines in your output. Printing each line correctly, in the correct order, 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
1x^2 + 4x + 3	Real 4 -1.000 -3.000
1x^2 + 1x + 1	Complex -3 -0.500+0.866i -0.500-0.866i
-2x^2 + 3x + 5	Real 49 -1.000 2.500
-20x^2 + 10000x + 50	Real 100004000 -0.005 500.005
-9x^2 + 24x + -16	Real 0 1.333 1.333
2x^2 + 4x + 16	Complex -112 -1.000+2.646i -1.000-2.646i

Grade Grab (p4v3d1)

Grade Grab [20 marks]

Problem Statement
Your instructor teaches another large course called CS771: Introduction to Machine Learning. Help him perform grading for that course. You will be given the total marks scored by a student out of 100, say M, the percentage of lectures the student has attended, say A, and a attendance weightage out of 100, say W. **All numbers will be given as an integer**.

The grade of a student is calculated by using their *effective marks*. If the marks of a student M are **greater than or equal to 50** then their effective marks is M itself. Otherwise, the effective marks is calculated as the **integer portion** of the following quantity (M + W% of A) i.e. by discarding digits after the decimal place. For example, if M = 33, A = 50 and W = 50 then effective marks will be (33 + 50% of 50) = (33 + 25) = 58. If the above calculation gives effective marks as **greater than 100**, then the effective marks is set to 100.

Given the effective marks of a student, grading is done using the following chart.

Marks	Grade
0-49	D
50-69	C
70-89	B
90-100	A

You have to give your output in four lines as shown below

1. In the first line, output the effective marks of the student **as an integer**
2. In the second line, print the grade the student got
3. In the third line, print the grade the student would have got if their marks were the same but attendance was 0%

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful that in the third line, the case may be such that the effective marks of the student changes due to change in attendance.
3. You may be tempted to use copy-paste while coding this question. Be careful while doing so to avoid making errors.

INPUT:
M A W

OUTPUT:
EffMarks
Grade
GradeZeroAttendance

EXAMPLE:
INPUT
83 50 50

OUTPUT:
83
B
B

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

There will be partial grading in this question. There are three lines in your output. Printing each line correctly, in the correct order, carries partial weightage. The first line is worth 20% weightage and the second

and third lines are worth 40% weightage each. 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
83 50 50	83 B B
33 50 50	58 C D
69 100 100	69 C C
40 85 82	100 A D
89 10 60	89 B B
38 64 50	70 B D