Homework: Java Syntax

This document defines homework assignments from the “Java Basics“ Course @ Software University. Please submit as homework a single **zip** / **rar** / **7z** archive holding the solutions (source code) of all below described problems.

Problem 1. Rectangle Area

Write a program that enters the **sides of a rectangle** (two integers **a** and **b**) and calculates and prints the rectangle's area. Examples:

Input	Output
7 20	140
5 12	60

Problem 2. Triangle Area

Write a program that enters 3 points in the plane (as integer **x** and **y** coordinates), calculates and prints the **area of the triangle** composed by these 3 points. Round the result to a whole number. In case the three points do not form a triangle, print "**0**" as result. Examples:

Input	Output

-5 10
25 30
60 15
	575

Input	Output

53 18
56 23
24 27
	86

Input	Output

1 1
2 2
3 3
	0

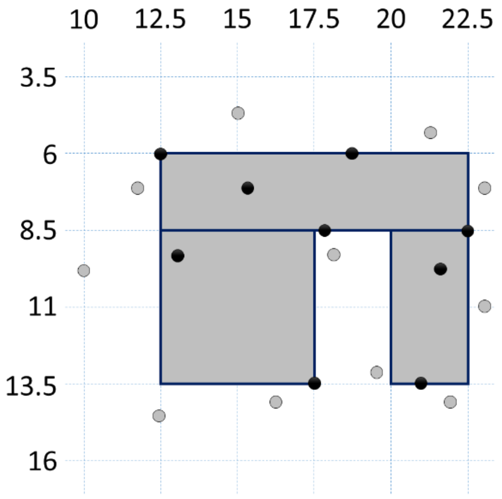
This resource could help you: http://www.mathopenref.com/coordtrianglearea.html.

Problem 3. Points inside a Figure

Write a program to check whether a point is **inside or outside of the figure** below. The point is given as a pair of floating-point numbers, separated by a space. Your program should print "Inside" or "Outside". Examples:

Input	Output
10 9.7	Outside
11.6 7	Outside
12.5 6	Inside
12.5 14.5	Outside
13.13 9.15	Inside
15.02 4.83	Outside
15.11 7.01	Inside
16.33 14.03	Outside
17.5 13.5	Inside
17.60 8.50	Inside

Input	Output
17.72 9.12	Outside
18.6 6	Inside
19.693 13.4	Outside
21 13.5	Inside
21.3 5.5	Outside
21.45 9.7	Inside
22 14	Outside
22.5 8.5	Inside
23 7.5	Outside
23.001 11.01	Outside



Problem 4. The Smallest of 3 Numbers

Write a program that finds the **smallest of three numbers**. Examples:

a	b	c	smallest
5	**2**	2	2
2	2	**1**	1
22	**4**	13	4
0	-2.5	**-5**	-5
-1.1	-0.5	-0.1	-1.1

Problem 5. Decimal to Hexadecimal

Write a program that enters a positive integer number **num** and converts and prints it in hexadecimal form. You may use some built-in method from the standard Java libraries. Examples:

Input	Output
254	FE
6779	1A7B

Problem 6. Formatting Numbers

Write a program that reads 3 numbers: an integer **a** (0 ≤ **a** ≤ 500), a floating-point **b** and a floating-point **c** and **prints them in 4 virtual columns** on the console. Each column should have a width of 10 characters. The number **a** should be printed in **hexadecimal, left aligned**; then the number **a** should be printed in binary form, padded with zeroes, then the number **b** should be **printed with 2 digits after the decimal point, right aligned**; the number **c** should be **printed with 3 digits after the decimal point, left aligned**. Examples:

a	b	c	result
254	11.6	0.5	|FE |0011111110| 11.60|0.500 |
499	-0.5559	10000	|1F3 |0111110011| -0.56|10000.000 |
0	3	-0.1234	|0 |0000000000| 3.00|-0.123 |
444	-7.5	7.5	|1BC |0110111100| -7.50|7.500 |

Problem 7. Count of Bits One

Write a program to **calculate the count of bits 1** in the binary representation of given integer number **n**. Examples:

n	binary representation of n	count
5	00000000 00000101	2
0	00000000 00000000	0
15	00000000 00001111	4
5343	00010100 11011111	9
62241	11110011 00100001	8
17263	01000011 01101111	9

Problem 8. * Count of Equal Bit Pairs

Write a program to count **how many sequences of two equal bits** ("**00**" or "**11**") can be found in the binary representation of given integer number **n** (with overlapping). Examples:

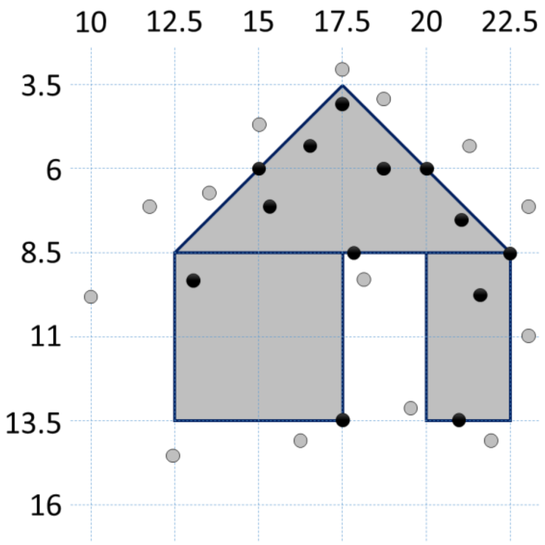
n	binary representation of n	count
5	101	0
0	0	0
15	1111	3
5343	1010011011111	6
62241	1111001100100001	9

Problem 9. ** Points inside the House

Write a program to check whether a point is **inside or outside the house** below. The point is given as a pair of floating-point numbers, separated by a space. Your program should print "Inside" or "Outside". Examples:

Input	Output
10 9.7	Outside
11.6 7	Outside
12.5 6	Outside
12.5 14.5	Outside
13.13 9.15	Inside
13.5 6.9	Outside
15 6	Inside
15.02 4.83	Outside
15.11 7.01	Inside
16.33 14.03	Outside
16.4 5.4	Inside
17.5 3	Outside
17.51 4.01	Inside
17.5 13.5	Inside

Input	Output
17.60 8.50	Inside
17.72 9.12	Outside
18.6 3.9	Outside
18.6 6	Inside
19.693 13.4	Outside
20 6	Inside
21 7.5	Inside
21 13.5	Inside
21.3 5.5	Outside
21.45 9.7	Inside
22 14	Outside
22.5 8.5	Inside
23 7.5	Outside
23.001 11.01	Outside



Hint: to check whether a point is inside a rectangle, you may check **at which side of a line a point lies**: http://stackoverflow.com/questions/1560492/ (for each of the triangle's sides the point should lie at the same side).

Problem 10. *** Paint a House as SVG

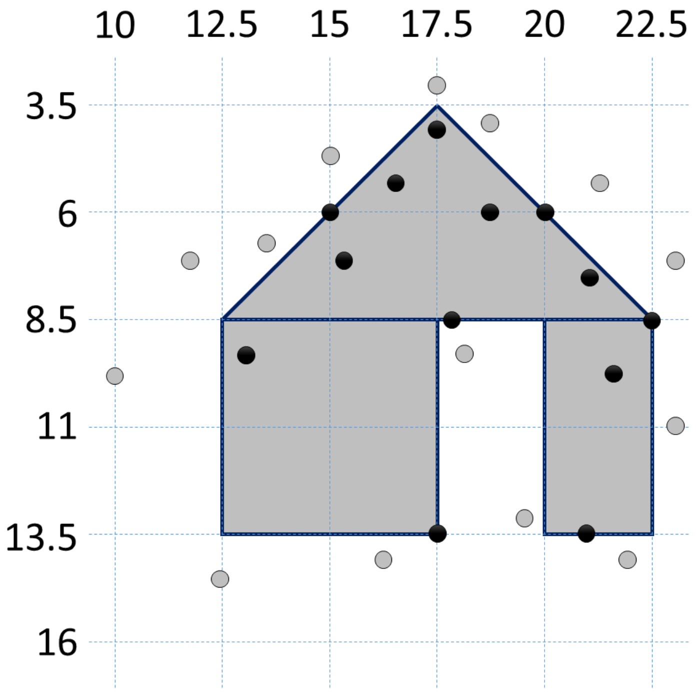
Write a program to **visualize the house and the points** from the image above as **SVG graphic embedded into a HTML** document. The SVG format (Scalable Vector Graphics) is a XML based format for describing vector graphics used in the modern Web applications that supports drawing lines, circles, ellipses, rectangles, paths and other objects like text and raster images. You may find in Internet some Java library to build SVG graphics or you may build it through an XML parser or by printing plain text. You are free to choose the libraries and tools.

- The output should look similar to the image above.
- The **coordinate axes** should be thin dotted lines. The coordinates should have **numbers** as above.
- The house should consist of **two rectangles and a triangle** above them, with solid lines and filled in semi-transparent gray color.
- The **points inside** the house should be painted as **black circles**.
- The **points outside** the house should be painted as **gray circles** with thin black border.

- You are not allowed to use ready SVG, use Java code to draw everything!

Take as **input** the coordinates of the points (the first line holds the number of points **n** and the next **n** lines hold a point coordinates separated by a space). Produce as **output** a file names **house.html**, which visualizes the house and the points inside and outside of it through an embedded SVG graphic. Example:

Input	Output

28
10 9.7
11.6 7
12.5 6
12.5 14.5
13.13 9.15
13.5 6.9
15 6
15.02 4.83
15.11 7.01
16.33 14.03
16.4 5.4
17.5 3
17.51 4.01
17.5 13.5
17.60 8.50
17.72 9.12
18.6 3.9
18.6 6
19.693 13.4
20 6
21 7.5
21 13.5
21.3 5.5
21.45 9.7
22 14
22.5 8.5
23 7.5
23.001 11.01
	house.html
	→
	

10
12.5
15
17.5
20
22.5
3.5
6
8.5
11
13.5
16