

Team \_\_\_\_\_  
Password \_\_\_\_\_

# CodeLM 2015

Questions

If you are in the advanced division, please skip ahead to problem 6.

### Problem 1: AB Rectangle

Given a width and height, print out a square that has a border of 'a's and is filled with 'b's. You will print the rectangle line by line and not return anything.

#### Sample Data

Input	Output
1, 1	a
2, 2	aa aa
3, 3	aaa aba aaa
4, 5	aaaa abba abba abba aaaa

## Problem 2: Combination Lock

You are attempting to crack a safe with a simple 6-digit combination lock. This combination lock has multiple answers according to the following pattern:

1. The first and fourth values must be both even or both odd.
2. The fifth and sixth values must add to the third value.
3. The second value must be at most 2 away from the third value.

Given a 6-digit guess, you must determine whether it matches the above pattern or not. Note that you can use  $\%10$  to get the far right number and  $/10$  to remove the far right number.

### Sample Data

Input	Output	Explanation
479236	true	All conditions are met.
197216	false	The first condition is not met.
357524	false	The second condition is not met.
639854	false	The third condition is not met.

### Problem 3: Tic-Tac-Toe Winner

Given a two-dimensional array of characters representing a valid tic-tac-toe board, you must determine which player, if either won. You will return the character of the winning player ('x' or 'o'), 't' if the game is a tie, or 'g' if the game is not yet over. There will never be a situation in which both players have won.

#### Sample Data

Input	Output	Explanation
<pre>{   { 'x', 'o', 'x' },   { 'o', 'x', 'o' },   { 'x', 'o', 'o' } }</pre>	'x'	X has a diagonal win.
<pre>{   { 'x', ' ', 'o' },   { ' ', ' ', 'o' },   { 'x', 'x', 'o' } }</pre>	'o'	O has a vertical win.
<pre>{   { 'x', 'o', 'x' },   { 'o', 'x', 'o' },   { 'o', 'x', 'o' } }</pre>	't'	The game is over and neither player won.
<pre>{   { ' ', 'o', ' ' },   { ' ', 'x', ' ' },   { 'o', 'x', ' ' } }</pre>	'g'	The board is not filled and neither player has won.

#### Problem 4: Magic Square

A magic square is a grid of numbers where all verticals, horizontals, and diagonals add to be the same number. The following is a 3x3 square where all lines sum to 15:

```
6 1 8
7 5 3
2 9 4
```

Given a two-dimension array of integers, return the number to which each line sums, or -1 if the input does not represent a magic square.

#### Sample Data

Input	Output	Explanation
<pre>{   { 6, 1, 8 },   { 7, 5, 3 },   { 2, 9, 4 } }</pre>	15	All lines sum to 15.
<pre>{   { 6 } }</pre>	6	There is only one number.
<pre>{   { 1, 4 },   { 2, 5 }, }</pre>	-1	$1 + 4 = 5$ , but $2 + 5 = 7$ .

### Problem 5: Odd Element Out

Given an array of integers, return a new array with all elements to the immediate right of odd numbers removed. Each odd number will only remove one other number. Note that removed odd numbers still count.

#### Sample Data

Input	Output	Explanation
{ 2, 4, 6 }	{ 2, 4, 6 }	There are no odd numbers.
{ 2, 3, 4 }	{ 2, 3 }	The number 4 is directly to the right of the number 3, which is odd.
{ 3, 5, 2 }	{ 3 }	5 is right of 3 and 2 is right of 5. Note that the 5 still removes the 2 even though it was already removed by the 3.

This is where the advanced division beings.

### Problem 6: Roman Numerals

Given a String that contains one number in roman numeral form, return that number as an int. You may assume that all numbers are valid. Here is a chart of roman numerals:

Roman	Decimal
I	1
V	5
X	10
L	50
C	100
D	500
M	1,000

### Sample Data

Input	Output
"I"	1
"IV"	4
"XXVI"	26
"DCXV"	615

### Problem 7: Time Formatter

Given a time in seconds, print a formatted time in hours, minutes, and seconds in the following format:

# hour(s) # minute(s) # second(s)

Where each # is replaced by the number of that unit. If a number is 0 (for example, less than one hour), omit that unit. If the number is 1, ensure that the unit is singular and not plural. The number 185 is “3 minutes 5 seconds” because there are 0 hours, 3 minutes, and 5 seconds.

#### Sample Data

Input	Output
90	“1 minute 30 seconds”
3600	“1 hour”
121	“2 minutes 1 second”
3662	“1 hour 1 minute 2 seconds”



## Problem 8: Special Substitution Cypher

A substitution cypher is an easy way to encrypt information. Every letter is replaced with the letter after it (a is replaced with b, b with c, etc.), and z is replaced with a. This cypher isn't terribly secure, so we will make a few modifications to it.

Given a message and an operation (either 'e' for encrypt or 'd' for decryption), you must encrypt or decrypt the given message. The steps below will encrypt the message. Following them in reverse will decrypt the message.

1. Reverse the string.
2. Remove all vowels from the string. They do not need to be added back in the decryption phase.
3. Replace all instances of a space with an underscore (\_).
4. Apply the substitution cypher as specified above.

### Sample Data

Input	Output	Explanation
"hello world", 'e'	"emxs_mmi"	<ol style="list-style-type: none"><li>1. "dlrow olleh"</li><li>2. "dlrw llh"</li><li>3. "dlrw_llh"</li><li>4. "emxs_mmi"</li></ol>
"emxs_mmi", 'd'	"hll wrld"	<ol style="list-style-type: none"><li>1. "dlrw_llh"</li><li>2. "dlrw llh"</li><li>3. Nothing</li><li>4. "hll wrld"</li></ol>

### Problem 9: Simple Calculator

Given an equation consisting of two numbers and an operator (addition, subtraction, multiplication, division, modulo, or power) you must compute the output. Return Double.NaN if the equation includes division or modulo by 0.

#### Sample Data

Input	Output
1 + 1	2
3 - 12	-9
3 * 4	12
81 / 9	9
10 % 3	1
2 ^ 3	8

## Problem 10: Array Operations

Given an operation and two arrays of ints (not necessarily of equal size), you must return the result of that operation in the form of an array. Note that the result could be an empty array. The operations are as follows:

**Union Operator ('u'):** Return an array that containing all values from both arrays with no repetition.

**Intersection Operator ('i'):** Returns an array containing all values that are in both arrays.

**Difference Operator ('d'):** Returns an array containing all values that are in the first array but not the second.

**Reverse Difference Operator ('r'):** Returns an array containing all values that are in the second array but not the first.

### Sample Data

Input	Output
'u', { 1, 2, 3 }, { 2, 3, 4, 5 }	{ 1, 2, 3, 4, 5 }
'i', { 1, 2, 3 }, { 2, 3, 4, 5 }	{ 2, 3 }
'd', { 1, 2, 3 }, { 2, 3, 4, 5 }	{ 1 }
'r', { 1, 2, 3 }, { 2, 3, 4, 5 }	{ 4, 5 }

If you are in the intermediate division, you are finished. Do not complete this problem.

### Problem 11: Binary-Decimal Converter

We normally count using the decimal system, where a number goes to the next place every power of 10. Computers count using the binary system, where a number goes to the next place at powers of 2.

Consider the decimal number 163. We could fill in a table to represent the number like so:

<b>Place</b>	100	10	1
<b>Count</b>	1	6	3

To convert this number to binary, we could fill in the table below:

<b>Place</b>	128	64	32	16	8	4	2	1
<b>Count</b>	1	0	1	0	0	0	1	1

To convert back to decimal, simply multiply each **count** by its **place** and add these numbers.

Knowing this information, you must now write a converter that takes in a number and the type of that number (“d” for decimal and “b” for binary) and converts it to the other type. You may assume that all numbers are valid and the largest number would be 11111111 or 255.

### Sample Data

<b>Input</b>	<b>Output</b>
163, 'd'	10100011
10100011, 'b'	163

## Problem 12: Safe Cracking Probability

You are attempting to crack a safe, but you aren't completely sure what the combination is. You will be given a String containing your guess. If you don't know a particular digit, you will see a guess in its place in the form [x,y] denoting that that particular digit is between x and y inclusive. You must compute the probability of cracking the safe given your guess and the correct answer. The guess will be the first parameter as a String. The answer will be the second parameter as an integer.

### Sample Data

Input	Output	Explanation
"12345", 12345	100	The guess supplied is correct.
"98765", 12345	0	The guess is wrong.
"12[0,9]45", 12345	10	All known digits are correct, and there is a 1 in 10 chance of the missing digit being correctly guessed.
"12[0,4]45", 12345	20	All known digits are correct, and there is a 1 in 5 chance of the missing digit being correctly guessed.
"12[0,1]4[0,1]", 12041	25	All known digits are correct, and there is a 1 in 2 chance of each missing digit being correctly guessed.
"12[0,1]4[0,1]", 12340	0	One of the guesses cannot be right.

### Problem 13: Special Sort

A Person class has been defined as follows:

<b>class</b>	Person
<b>variables</b>	private String name private int age
<b>methods</b>	public String getName() public int getAge()

Given an ArrayList of type Person, sort the ArrayList by age (youngest to oldest) then by name (A-Z).

#### Sample Data

Input	Output	Explanation
[ Person("Noah", 16), Person("Eli", 15) ]	[ Person("Eli", 15), Person("Noah", 16) ]	Noah and Eli are swapped for age.
[ Person("David", 16), Person("Noah", 16) ]	[ Person("David", 16), Person("Noah", 16) ]	The ages are the same and the names are already sorted.
[ Person("Noah", 16), Person("Nathan", 16) ]	[ Person("Nathan", 16), Person("Noah", 16) ]	The ages are the same and Noah must swap with Nathan.
[ Person("Noah", 16), Person("Max", 16), Person("Gabe", 15), Person("Dan", 18) ]	[ Person("Gabe", 15), Person("Max", 16), Person("Noah", 16), Person("Dan", 18) ]	Sorting by age produces the following:  [ Person("Gabe", 15), Person("Noah", 16), Person("Max", 16), Person("Dan", 18) ]  We must now swap Noah and Max to put their names in order.

### Problem 14: Shipment Order

A shipment-tracking program has recently broken and needs to be fixed. However, the program is poorly written, so it won't be as easy as you think.

You will be given three arrays of equal length representing the shipment data. The first array will contain the items being shipped. The second array will contain the times in 12-hour format. The last array will tell you whether each time is am or pm. Your task is to return an array containing the items being shipped in order from earliest to latest.

#### Sample Data

Input	Output
{ "iPhone 6S", "MacBook Ultra", "iPad Pro" }, { 1250, 609, 900 }, { "pm", "am", "pm" }	{ "MacBook Ultra", "iPad Pro", "iPhone 6S" }

### Problem 15: Maze Solver

You will be given a maze like this:

```
o——
o—o—o
—oo—o
—o—oo
```

Your task is to determine, starting at (0, 0), whether the maze can be solved. The maze is solvable if each o has at least one neighbor (vertical, horizontal, diagonal) and that moving from o to o will eventually land you at the bottom right of the array. Note that if (0, 0) is not an o, the maze is unsolvable. Return true if the maze is solvable, false if not.

#### Sample Data

Input	Output	Explanation
<pre>{   "o——",   "o—o—o",   "—oo—o",   "—o—oo" }</pre>	true	One possible solution is:  (0, 0) (0, 1) (1, 2) (2, 2) (3, 3) (4, 3)