CodeLM 2017

Expert Division

School Name

Dashboard Login Information:

Username: x

Password: x

Network Login Information:

SSID: lmsdcode

Password: lmsdcode

Rules: You will have until 11 o'clock complete your division's six questions. All questions must be submitted through the website codeLM.com and have a runtime of less than 10 seconds. You will submit your code by copying and pasting the code from your IDE to each problem's textbox on the website and then hitting submit.

Starter code should be downloaded from codeLM.com. **Do not change the names of the classes in this starter code or method headers.** If you do your code will not compile.

Problems 1 and 2 are each worth 2 points; Problems 3 and 4 are each worth 3 points; Problems 5 and 6 are worth 4 points. In order to receive credit for these questions your code must produce the correct results for all test data. This test data will not be the same as the sample data that is provided with each question in this packet, so be sure to thoroughly test your code before you submit it. Incorrect submissions that compile will result in the loss of one point. If your submission does not compile you will not lose a point. If you believe that your code has incorrectly been marked as wrong you should inform a judge who will then have the question reviewed. You may make as many submissions for each question as you would like, although you will only receive credit for one correct answer per problem.

Important Things to Remember

- You must download the starter code from codeLM.com. This starter code contains code that will read in all values that will be necessary to solve each problem from standard input.
- Do not change the class names in the starter code. If you do your code will not compile.
- Your code should not print out anything except for the answer. Your answer must be formatted exactly like the sample data on codeML.com or it will be marked wrong.

Index:

Problem 1: Recalled Hardware

New Wave Computer's™ new product, the Universe Edge, has had some minor "malfunctions". Specifically, their phones' batteries have been know to charge slowly and spontaneously combust. To avoid legal fallout, certain Universe Edge units have to be recalled. You are to design a program that receives a serial number and determines if the phone is faulty.

Input
String **`serialNumber`**: The serial number is eleven characters long.

- Characters 1, 2 and 3 represent the factory code. The first two characters are letters which correspond to the state's abbreviation. New Wave has factories in the states of PA, NY, NJ and AK. The third character is an integer which denotes the factory number in the specific state.
- Characters 4 and 5 are integers and represent the month of manufacture.
- Characters 6 and 7 are integers and represent the last two digits of the year of manufacture.
- Characters 8, 9, 10 and 11 form the unique identifier for each phone. The identifier is a mixture of letters and numbers. To form the integer UID, convert all letters into their corresponding ASCII values, then concatenate these ascii values, together with the original digits. For example, the UID was 9aD7 would evaluate to

  9 + 97 + 68 + 7

  Which would be concatenated to form

  997687

  UID's can not start with a zero.

Output
A boolean which is true if the phone should be recalled and is false otherwise. The following factors should be used to determine whether or not a phone is faulty:

- If the phone was manufactured in West Chester, Pennsylvania, which has a factory number 6, it is faulty.
- Phones manufactured from November 2015 to February 2016 in New Jersey are also faulty.
- All phones manufactured in October of 2016 in New York or Alaska are faulty unless the unique identifier is divisible by 9, but not by 27.
- Regardless of the above guidelines, if the unique identifier is a palindrome, the phone is not faulty. A palindrome is a series of characters that is the same when reversed (eg. racecar = racecar, 1221 = 1221).

Sample Data

Input	Output	Explanation
`serialNumber` `= PA601162o12`	false	The unique identifier becomes 211112, which is a palindrome

serialNumber = PA60811a9bc	true	The phone was manufactured in West Chester, PA
serialNumber = NJ20116yy35	true	The phone was manufactured in New Jersey between November 2015 and February 2016

Problem 2: Newmerals

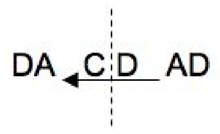
In order to improve security, New Wave Computers™ wants to implement a proprietary number system named Newmerals™. This system consists of four symbols: A, B, C, and D, where A < B, B < C and C < D. This system is so secretive that Newmerals cannot be translated into base-10 numbers. Newmerals can be combined using three operators: join, which is represented by a colon ':', promotion, which is represented by a carrot '^', and mirror, which is represented by a percent symbol '%'.

Input
String **`newmeralA`**, **`newmeralB`**: The two Newmerals. **The length of the longer Newmeral will be divisible .**

char **`operator`**: The operation that will be performed on the two Newmerals. This char can have one of three values: ':', '^', or '%'.

Output
A string that contains the Newmeral that will result from **`newmeralA`** combined with **`newmeralB`** using **`operator`**. Each of the three operations are described in the table below.

Operator	Function
Join ':'	Split each Newmeral into *n* groups, where *n* is the length of the shorter Newmeral. Join the groups, alternating between the two Newmerals, starting with the first. For example, if the two Newmerals had a length of two, you would combine them by selecting the first digit from the first Newmeral, followed by the first digit from the second Newmeral, then the second digit from the first Newmeral and finally the second digit from the second Newmeral. This operation results in a four digit Newmeral, as shown below: **AB** : CD = **A**C**B**D If the two Newmerals are of different lengths, the longer Newmeral will always be divisible by the shorter Newmeral. In the resultants of cases below, each character of the shorter Newmeral is separated by 2 characters of the longer Newmerals. In these cases, the shorter Newmerals' characters are separated by 2 because the length of the longer Newmeral is greater than the length of the smaller Newmeral by a factor of two. **AB** : DCBA = **A**DC**B**BA **DCBA** : AB = **DC**A**BA**B In the below case, the longer Newmeral is longer by a factor of 3 so the shorter Newmeral's character are each separated by 3 of the longer Newmeral's characters. **AB** : DCADBA = **A**DCA**B**DBA

	(Digits are bolded and underlined for demonstrative purposes only.)
Promotion '^'	Compare the two digits in each place value from the two Newmerals and construct a new Newmeral that is composed of the higher digit in each place value. Always start from the right and work to the left. For example, ABC ^ CDB = CDC You would evaluate this expression by comparing the C from the first Newmeral with the B from the second Newmeral and selecting C. You would then compare the B from the first Newmeral to the D in the second Newmeral and select D. Finally, you would compare the A from the first Newmeral with the C from the second and select C. **If the Newmerals are of unequal length, simply concatenate the remaining digits from the longer Newmeral.** If two digits are the same, promote that digit by one (i.e. two C's would become D, etc.) If the two digits are both Ds, they should overflow and become an A. For example, AD ^ AD = BA In this example, the two D's would be compared to each other and be promoted to an A. The two A's would also be compared to each other and be promoted to B since they are the same.
Mirror '%'	The Mirror operator reflects the second Newmeral over the first Newmeral. For example, CD % AD = DACDAD In this example AD is reflected over CD which would give you DACD. AD then follows CD. 

Sample Data

Input	Output	Explanation
newmeralA = ACD **newmeralB** = BDBCDD operator = :	ABDCBCDDD	The 'join' operation is performed on the two Newmerals.
newmeralA = AD **newmeralB** = CD **operator** = ^	CA	The 'promotion' operation is performed on the two Newmerals. The operation is evaluated by comparing the A from the first Newmeral with the C from the second Newmeral and selecting C. You would then compare the D from the first Newmeral to

		the D in the second Newmeral. Since they are both the same Newmeral you would promote one, which in this case would wrap around to A.
`newmeralA` = AC **`newmeralB`** = DC **`operator`** = %	CDACDC	The converse of the second Newmeral, “DC” is placed in front the first to Newmeral to get CDAC. The second Nemeral is then appended to the the end of this string.

Problem 3: nCrypt

New Wave has decided to encrypt all of their communications so that attackers cannot steal their trade secrets. They will use a highly advanced encryption technique in which each letter is replaced by the letter **x** letters after it in the alphabet. The message can include uppercase and lowercase letters. Any character that is not a letter should not be changed.

Using this algorithm, the word “The” encrypted with an offset of 4 would become “Xli”, because ‘X’ is four letters after ‘T’ in the alphabet, ‘l’ is four letters after ‘h’ and ‘i’ is four letters after ‘e’.

If you were to attempt to encrypt a letter, but when counting **x** letters after that letter you find that you are outside of the alphabet, you should wrap around to the beginning of the alphabet. For example, if you were to encrypt the letter ‘y’ with an offset of 3, you would get the letter ‘b’.

Many employees are not used to the new system, so they encode their messages without remembering the rotation factor. Your task is the find the most probable **x** in which the original message was encoded. Certain words are very common in New Wave transmissions, so they can indicate a correct rotation factor. To determine the best rotation factor, score each permutation by adding the length of each common word every time it appears in the decoded message (it is case sensitive). The list of common words is below:

NWPC	Rubin	or	DDOS
New	myPhone	I	spontaneous
Wave	smudged	duct	combustion
Noah	and	tape	Newmeral

In addition, important words in sequence can multiply the score. If two words appear back-to-back, their combined score should be doubled. If three words appear in a row, their combined score should be tripled. Continue the pattern for all *n*. Finally, if a word has an apostrophe-s or s, but the root word appears in the list, it should still count. Finally, a word that is followed by a period or a comma should still be counted.

Input
String `message`: The encrypted message.

Output
The most probable value of **x**.

Sample Data

Input	Output	Explanation
`message` `= Opbi Svcjo hpu EEPTfe`	1	When this message is encrypted with offset of 1 it becomes “Noah Rubin got DDOSed”. “Noah” has a score of 4, “Rubin” has a score of 5, since these words are consecutive, each of their scores would be doubled. DDOS appears in the sentence, however it should not be counted because it has an “ed” added to it
`message` `= Uovigf hygx xeti.`	4	Encrypting this message with an offset of 4 would give you “Qkrecb duct tape.”, which

		would give a score of 16. Note that the period does not affect the score. Moving each letter 2 spaces would result in “smudged fwev vcrg”, which would have a score of 7.
`message` `=  Ofx Xbwf Pqcj abcd Twdkp`	1	Offset of 1= New Wave…….. Offset of two =..... Noah yzab Rubin. Offset of one scores more because its score is multiplied by two for a total of 14 points. Offset of one only scores 9 points.

Problem 4: Vending Machine

The vending machines in the New Wave headquarters are acting up. Entering a combination will sometimes yield the wrong snack. Other times it will follow a path that results in a dead end with no output. Occasionally the machine will even go into an infinite loop and crash.

Fortunately, the vending machine does print a stack-trace that shows what choice was made and the series of paths that were taken by the machine, although these paths are not usually in the correct order. Each individual path consists of a starting point and an ending point. The starting point will always be a two character combination of a letter between A and F, inclusive, and a digit from 0-9, inclusive. The ending point will either also be a two character combination as previously specified, or a snack.

For example, when “B1” is entered the machine returned the series:

{ {“C3”, “A5”}, {“B1”, “C3”}, {“A5”, “Crackers”} }, “B1”

indicating that the user selected “B1” and three jumps were made by the machine, where C3→A5, B1→C3, and A5→Crackers. These connections should be rearranged to show that entering B1 would result in the purchase of Crackers:

B1→C3
C3→A5
A5→Crackers

You are to write a program that will take the stack-trace and determine what the result will be, whether it is a snack, dead end, or an infinite loop.

Input
String `start`: A string with a length of two. The first character in this String will be a letter between A and F, inclusive. The second will be a digit from 0-9, inclusive.

An array of size-2 arrays `path`: The first element in each size-2 array is a String that denotes the starting point for a connection. The first character in this String will be a letter between A and F, inclusive. The second will be a digit from 0-9, inclusive. The second element is what that starting point leads to. This can be either a two character String just like the starting point or a snack. You should assume that this second element is a snack if it is not a valid element.

Output
A String that indicates either the snack the customer will get, “Nothing” if the path leads to a dead end, or “Loop” if the machine enters an infinite loop.

Sample Data

Input	2d Array Representation of path	Output	Explanation
start = A6 **path** = E5 D4 D4 B2 F5 C3 A6 B9 C4 D6 B9 F5 B2 E5 C3 PopTart	{{“E5”, “D4”}, {“D4”, “B2”}, {“F5”, “C3”}, {“A6”, “B9”}, {“C4”, “D6”}, {“B9”, “F5”},	“PopTart”	Vending machine follows the path: A6 → B9 → F5 → C3 → PopTart

	`{“B2”, “E5”},`
	`{“C3”, “PopTart”}}`		

`start` `= C4`
`path` `= E5 D4 D4 B2 F5 C3 A6 B9 C4 D6 B9 F5 B2 E5 C3 PopTart`
	`{{“E5”, “D4”},`
	`{“D4”, “B2”},`
	`{“F5,  “C3”},`
	`{“A6”, “B9”},`
	`{“C4”, “D6”},`
	`{“B9”, “F5”},`
	`{“B2”, “E5”},`
	`{“C3”, “PopTart”}}`
		“Nothing”	Program follows the path:
			C4 → D6
			prints “Nothing”

`start` `= B2`
`path` `= E5 D4 D4 B2 F5 C3 A6 B9 C4 D6 B9 F5 B2 E5 C3 Poptart`
	`{{“E5”, “D4”},`
	`{“D4”, “B2”},`
	`{“F5,  “C3”},`
	`{“A6”, “B9”},`
	`{“C4”, “D6”},`
	`{“B9”, “F5”},`
	`{“B2”, “E5”},`
	`{“C3”, “PopTart”}}`
		“Loop”	Program follows the path:
			B2 → E5 → D4 → B2 → E5…
			prints “Loop”

Problem 5: **Newber**

New Wave's automated car service, Newber, has received reports that the navigation system in their cars has come under attack. The cars need to be programmed to drive back to a designated station for recalibration.

Input

A two-dimensional 4x4 array of chars, **`map`**, showing the current location of the car and the status of the terrain surrounding it. This 4x4 array will be composed from a collection of the following four characters:

- 'X'   - The starting position of the car. This starting point can be anywhere in the array. There will only be one starting point.
- 'O' - The station endpoint. This endpoint can be anywhere in the array. There will only be one ending point.
- '-'   - An empty street
- '|'   - An obstacle which must be driven around.

Output

A string value of 'R's, 'L's, 'U's and 'D's such that

- The *string* represents the shortest path from start to finish
- Each 'R' represents a move to the right, 'L' a move to the left, 'U' a move upwards, and 'D' a move downwards
- Moves are only in these four directions and not diagonal
- Should the path be traced starting from the 'X,' the car would only travel across '-' characters, never pass through '|' characters and end at the 'O' character
- There will never be a tie for shortest route and there will always be a route to the finish.

Sample Data

Input	Output	Explanation

`map=`
`X - | -`
`- | - -`
`- - | O`
`| - - -`
	DDRDRRU	To get from the X to the O you would have to go down-down-right-down-right-right-up.

`map=`
`- - - -`
`| - | -`
`| X | O`
`| - - -`
	DRRU	There are two paths that can get you to the O. down-right-right-up is the shortest path.

`map=`
`0 - - -`
`| - | -`
`| | | X`
`| - - -`
	UULLL	The path from the X to the 0 is up-up-left-left-left.

Problem 6: Mole in the Company

Secret messages passed to a competing company have recently been intercepted. It is unknown which employee is sending these messages, but every message ends with a **n** x **n** sequence of numbers, which is clearly an obfuscated ID number. Because of the mole’s arrogance, New Wave was able to uncover the algorithm used to encode the ID number. New Wave needs your help to implement the algorithm and find the mole.

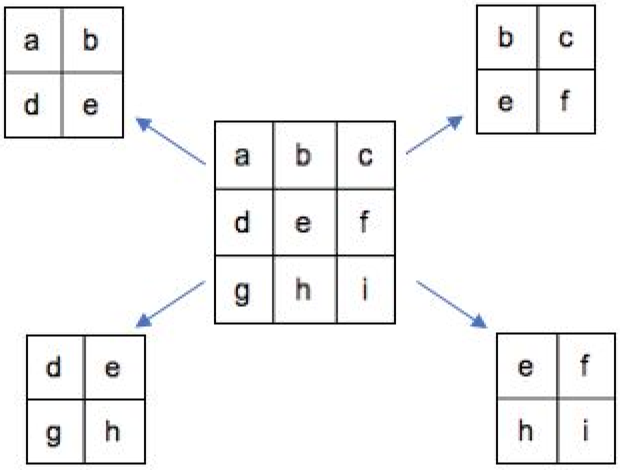
The numbers are always found in square combinations, and can be evaluated using the following algorithm:

The simplest sequence of numbers is a **2x2** grid.

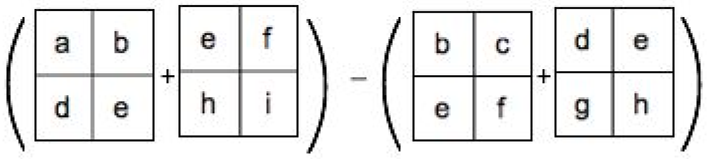
a	b
c	d

In this case, the ID number can be found by using the expression (a+d) - (b+c).

A **3x3** grid would be decomposed into four 2x2 grids, one for each corner of the square.



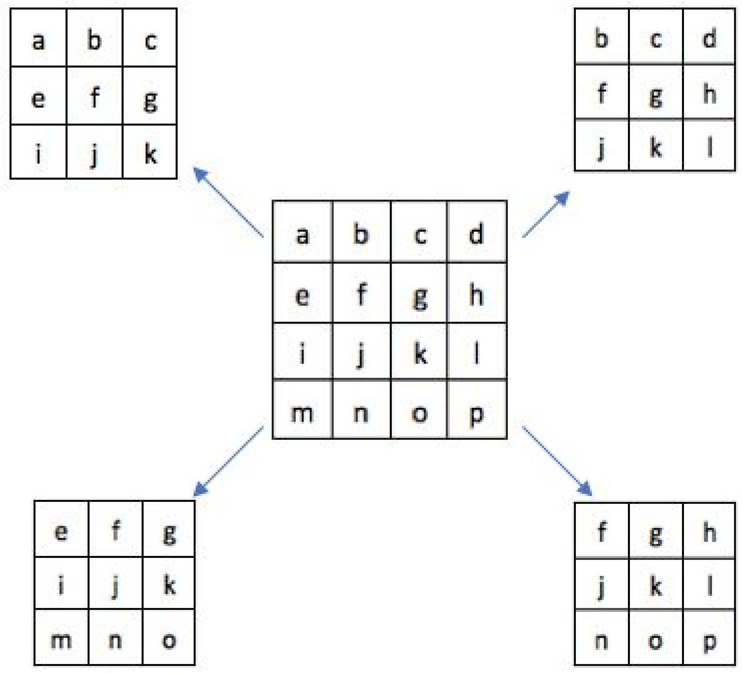
Once broken down into four 2x2 grids, each individual grid would be evaluated in the same manner as the original 2x2 grid that was demonstrated above. These four numbers would also be evaluated in the same manner as the original 2x2, so that you would get:



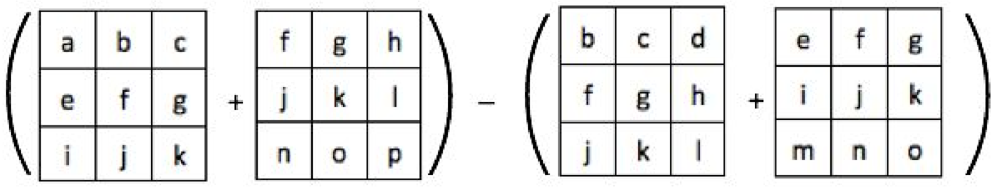
or

(((a+e) - (b+d)) + ((e+i) - (f+h))) - (((b+f) - (c+e)) + ((d+h) - (e+g)))

A **4x4** grid would be decomposed into four 3x3 grids, one for each of its corners.



Each of the four 3x3 grids would be evaluated just like the one in the example above, and then these four grids would themselves be evaluated using the same pattern so that the 4x4 grid would evaluate to:



This pattern continues for all nxn grids.

Input
code: An nxn array/list of integers.

Output
An integer that is equal to the deciphered value of the nxn array **code**.

Sample Data

Input	Output	Explanation

`code =`
`1 4`
`4 0`
	`-7`	Simple 2 by 2 matrix. Answer is just AD-BC or (1+0) -(4+4)

`code =`
`1 2 3`
	`0`	When divided into individual 2 x 2 matrices they are equal to -3,-3,-3,3. When you use the AD-BC equation you get (-3+-3)

4 5 6 7 8 9		-(-3+3) which is 0
code = 1 3 5 6 7 2 9 1 4 2 5 6 1 2 8 9 2 3 4 1 5 0 2 1 5	94	
code = 2 3 1 0 3 4 7 1 3 4 5 6 7 2 9 0 1 8 2 3 4 1 2 3 4 5 6 7 2 3 4 5 6 7 8 1 8 0 2 3 2 1 2 1 3 4 5 2 1	-1388	

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	