Instructions for students:

- Complete the following methods on Arrays
- ONLY USE NUMPY ARRAY. DO NOT USE LIST.
- You may use any language to complete the tasks.
- All your methods must be written in one single .java or .py or .pynb file. DO NOT CREATE separate files for each task.
- If you are using JAVA, you must include the main method as well which should test your other methods and print the outputs according to the tasks.
- If you are using PYTHON, then follow the coding templates shared in this folder.

NOTE:

- **YOU CANNOT USE ANY BUILT-IN FUNCTION EXCEPT len IN PYTHON. [negative indexing, append is prohibited]**
- **YOU HAVE TO MENTION SIZE OF ARRAY WHILE INITIALIZATION**
- **YOUR CODE SHOULD WORK FOR ANY VALID INPUTS. [Make changes to the Sample Inputs and check whether your program works correctly]**
- **Use Numpy array to initialize the array. You are allowed to use the shape parameter of the numpy array as well.**
- **DO NOT USE DICTIONARY**

Linear Array + 2D Array [CO5]

[Each method carries 5 marks]

1. ## Merge Lineup:

Now you guys are playing a 2v2 fierce pokemon battle. At one point your and your teammate's pokemons have very low hp. So you created a new rule to merge the hp of your and your teammate's pokemons and create a new pokemon lineup. But the opposite team placed a special condition. You and your teammate have to add the hp of your team's pokemons from opposite directions. That is, your first pokemon's hp will be added to your teammate's last pokemon's hp; your second pokemon's hp will be added to your teammate's second last pokemon's hp and so on and so forth. The None elements denote 0 hp. You and your teammate have the same number of pokemons. Implement the scenario using a proper method which takes your and your teammate's pokemon hp as two arrays (as parameters); and returns the resulting arr.

Sample Input / Driver Code:
pokemon_1: [4, 5, -1, None, None]
pokemon_2: [2, 27, 7, 12, None]

#Function Call: print(mergeLineup(pokemon_1, pokemon_2))

Sample Output:
[4,17,6,27,2]

Explanation:
4+None(0) = 4 {pokemon_1[0]+pokemon_2[4]},
5+12 = 17 {pokemon_1[1]+pokemon_2[3]},
-1+7 = 6 {pokemon_1[2]+pokemon_2[2]},
None(0)+27 = 27 {pokemon_1[3]+pokemon_2[1]},
None(0)+2 = 2 {pokemon_1[4]+pokemon_2[0]}

2. Discard Cards:

You and your friends are playing a card game. You pick n numbers of UNO cards from the deck. The rule of the card game is- one person says a number, **t** and you have to discard the **alternative** cards with the same number from your hand without changing the relative position of other cards. Implement the scenario using a proper method which takes the cards in your hands as an array and a number as parameters; and returns the resulting array. The empty spaces are denoted as 0 in the array.
Start from deleting the first t.

Sample Input / Driver Code:
cards=[1,3,7,2,5,2,2,2,0]
 	
#Function Call:
print(discardCards(cards, t = 2))
 	
Sample Output:
[1,3,7,5,2,2,0,0,0]
 	
Explanation:
[1,3,7,2_0,5,2_1,2_2,2_3,0]
Starting from the first 2, the alternative 2's are 2_0 and 2_2.
After removing them, the resulting array is -
[1,3,7,5,2,2,0,0,0]
	Sample Input / Driver Code:
	cards=[5,5,5,0,0]
	
	#Function Call:
	print(discardCards(cards, t = 5))
	
	Sample Output:
	[5,0,0,0,0]
	
	Explanation:
	[5_0, 5_1, 5_2,0,0]
	Starting from the first 5, the alternative 5's are 5_0 and 5_2.
	After removing them, the resulting array is -
	[5,0,0,0,0]

3. Decryption Process:

Suppose you're working as a cryptographer for a secret intelligence agency. You are given an encrypted matrix as an input. You have to decrypt it after some processing of the given matrix.

To decrypt this message efficiently, you have a task to construct a method called **decrypt_matrix(matrix)** which takes an encrypted matrix as input and returns a decrypted linear array. The process of finding the decrypted linear array is given below:

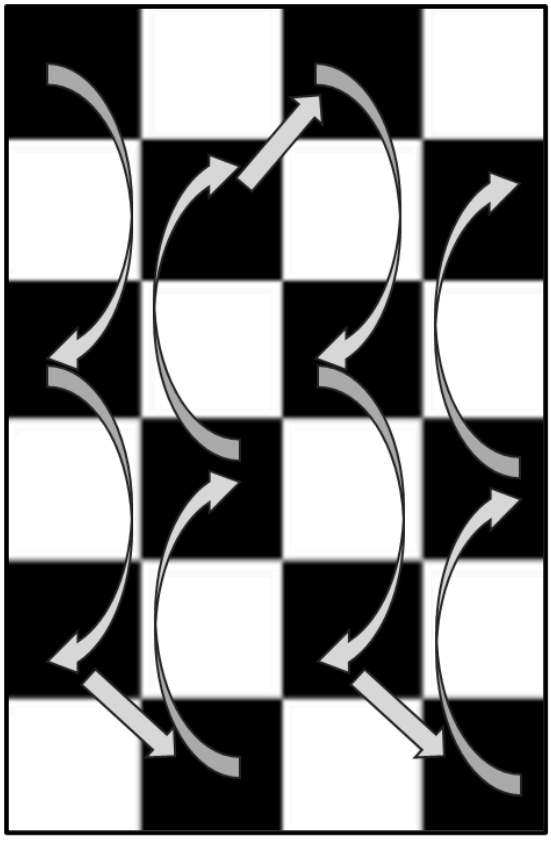
You have to find out the column-wise summations for each column and store the difference of subsequent column-wise summations in a new linear array.

<table>
<tr><th>Sample Input</th><th>Sample output</th><th>Explanation</th></tr>
<tr>
<td>
<table>
<tr><td>1</td><td>3</td><td>1</td></tr>
<tr><td>6</td><td>4</td><td>2</td></tr>
<tr><td>5</td><td>1</td><td>7</td></tr>
<tr><td>9</td><td>3</td><td>3</td></tr>
<tr><td>8</td><td>5</td><td>4</td></tr>
</table>
</td>
<td>
<table>
<tr><td>-13</td><td>1</td></tr>
</table>
</td>
<td>
Sum of 0th column = 29
Sum of 1st column = 16
Sum of 2nd column = 17

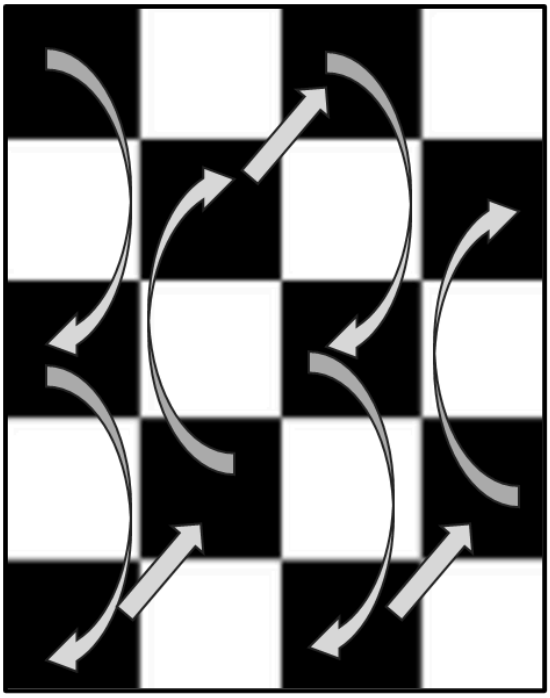
Therefore, the size of the resulting array is 2 and the array is:
<table>
<tr><td>16-29 = -13</td><td>17-16 = 1</td></tr>
</table>
</td>
</tr>
</table>

4. Zigzag Walk:

As a child, you often played this game while walking on a tiled floor. You walked avoiding a certain color, for example white tiles (it almost felt like if you stepped on a white tile, you would die!). Now you are in a room of m x n dimension. The room has m*n black and white tiles. You step on the black tiles only. Your movement is like this:



OR



Now suppose you are given a 2D array which resembles the tiled floor. Each tile has a number. Can you write a method that will print your walking sequence on the floor?

Constraint: The first tile of the room is always black.

Hint: Look out for the number of rows in the matrix [Notice the transition from column 0 to column 1 in the above figures]

Sample Input	Sample Output
---------- | 3 | 8 | 4 | 6 | 1 | ---------- | 7 | 2 | 1 | 9 | 3 | ---------- | 9 | 0 | 7 | 5 | 8 | ---------- | 2 | 1 | 3 | 4 | 0 | ---------- | 1 | 4 | 2 | 8 | 6 | ----------	3 9 1 1 2 4 7 2 4 9 1 8 6
---------- | 3 | 8 | 4 | 6 | 1 | ---------- | 7 | 2 | 1 | 9 | 3 | ---------- | 9 | 0 | 7 | 5 | 8 | ---------- | 2 | 1 | 3 | 4 | 0 | ----------	3 9 1 2 4 7 4 9 1 8

5. Row Rotation Policy of BRACU Classroom:

You are no longer permitted to choose your own seat on the new campus of BRACU. You must abide by the new regulations, which state that if you sit in the first row for the first week, you must shift to the second row for Week2, the third row for the week3, and so on. In your classroom, there are a total of 6 rows and 5 columns. Your friend "AA" wants to know from you that on the upcoming exam week in which row he will be in. Your task is to implement a function **row_rotation(exam_week, seat_status)** that takes the exam_week and a 2D array of current seat status as input and returns the row number in which your friend "AA" will be seated and print the seat status for that week.

Input: exam_week = 3
current_seat_status=

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y
Z	AA	BB	CC	DD

Output:

U	V	W	X	Y
Z	AA	BB	CC	DD
A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T

Your friend AA will be on row 2.

Explanation: exam is 3 week away from the current week
Current seat status of above input is for this week.

Seat Status of Next week:

Z	AA	BB	CC	DD
A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Seat Status of Exam week:

U	V	W	X	Y
Z	AA	BB	CC	DD
A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T

6. Matrix Compression:

Complete the function **compress_matrix** that takes a 2D array as a parameter and return a new compressed 2D array. In the given array the number of row and column will always be even. **Compressing a matrix means grouping elements in 2x2 blocks and sums the elements within each block. Check the sample input output for further clarification.**

Hint: Generally the block consists of the (i,j), (i+1,j), (i,j+1) and (i+1, j+1) elements for 2x2 blocks.

You cannot use any built-in function except len() and range(). You can use the np variable to create an array.

Python Notation	Java Notation

import numpy as np
def compress_matrix (mat):
    # To Do
	public int[ ][ ] compress_matrix (int[] [] mat) {
	    // To Do
	}

Sample Input array	All Box (No need to create these arrays)	Returned Array	Explanation
[[1, 2, 3, 4], [5, 6, 7, 8], [1, 3, 5, 2], [-2, 0, 6,-3]]	[[1, 2], [[3, 4], [5, 6]] [[7, 8]] [[1, 3], [[5, 2], [-2, 0]] [[6, -3]]	[[14, 22], [2, 10]]	[[1+2+5+6, 3+4+7+8], [1+3+-2+0, 5+2+6+-3]]

7. Game Arena:

Suppose you and your friends are in the world of '*Alice in Borderland*' where you decided to take part in a game and entered the *game arena*. As a team, you need to gain **at least 10 points** in order to keep surviving in the borderland. Otherwise, you will be out of the game and your team will be banished for good. Now, the arena has a 2D array like structure where players of a team are given certain positions with values that are **multiples of 50**. By staying in these positions, every player can gain points from the cells above, below, left and right (not diagonally) only if those cells **contain 2** [The cells containing 1s and 0s are to be avoided]. For each player, add from these cells containing 2s to your total points for the team to keep on surviving in borderland. **Be careful about corner cases. Your task is to write a method which tells us whether your team is out or has survived the game.**

Sample Input 1	Sample Output 2
---------- **| 0 | 2 | 2 | 0 |** **----------** **| 50 | 1 | 2 | 0 |** **----------** **| 2 | 2 | 2 | 0 |** **----------** **| 1 | 100 | 2 | 0 |** **----------**	**Points Gained: 6. Your team is out.**

Explanation:
Player with value 50 has 2 in the cell below him (1 cell). Player with value 100 has 2 in the cell above and in the right cell (2 cells). So in total, they got (1+2)*2 = 6 points which was not enough to survive the game.

Sample Input 1	Sample Output 2
---------- **| 0 | 2 | 2 | 0 | 2 |** **----------** **| 1 | 50 | 2 | 1 | 100 |** **----------** **| 2 | 2 | 2 | 0 | 2 |** **----------** **| 0 | 200 | 2 | 0 | 0 |** **----------**	**Points Gained: 14. Your team has survived the game.**

Explanation:
Player with value 50 has 2 in the cell above, cell below and the right cell (3 cells). Player with value 100 has 2 in the cell above and the cell below (2 cells). Player with value 200 has 2 in the above cell and right cell (2 cells). So in total, they gained (3+2+2)*2 = 14 points and survived the game.

Note: For the cell with value 2 that is common between 2 players in position (2,1), both gained 2 points each, so it's not like if one player already added those 2 points, another player cannot.