

## Multi-Dimensional Array related problems (Total 15 questions)

SL	Problem statement	Difficulty levels						
1.	<p>WAP that will take 9 integers into a 3 by 3 array (2D) and show them as traditional matrix view.</p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>9 8 7 6 5 4 3 2 1</td><td>9 8 7 6 5 4 3 2 1</td></tr><tr><td>1 1 1 2 2 2 3 3 3</td><td>1 1 1 2 2 2 3 3 3</td></tr></table>	Sample input	Sample output	9 8 7 6 5 4 3 2 1	9 8 7 6 5 4 3 2 1	1 1 1 2 2 2 3 3 3	1 1 1 2 2 2 3 3 3	*
Sample input	Sample output							
9 8 7 6 5 4 3 2 1	9 8 7 6 5 4 3 2 1							
1 1 1 2 2 2 3 3 3	1 1 1 2 2 2 3 3 3							
2.	<p>WAP that will take (m x n) integers into a <i>m</i> by <i>n</i> array (2D) and print them both row-wise and column-wise.</p> <table><tr><th>Sample input (m,n)</th><th>Sample output</th></tr><tr><td>2 3 1 2 3 6 5 4</td><td>Row-wise: 1 2 3 6 5 4 Column-wise: 1 6 2 5 3 4</td></tr><tr><td>3 3 1 1 1 2 2 2 3 3 3</td><td>Row-wise: 1 1 1 2 2 2 3 3 3 Column-wise: 1 2 3 1 2 3 1 2 3</td></tr></table>	Sample input (m,n)	Sample output	2 3 1 2 3 6 5 4	Row-wise: 1 2 3 6 5 4 Column-wise: 1 6 2 5 3 4	3 3 1 1 1 2 2 2 3 3 3	Row-wise: 1 1 1 2 2 2 3 3 3 Column-wise: 1 2 3 1 2 3 1 2 3	*
Sample input (m,n)	Sample output							
2 3 1 2 3 6 5 4	Row-wise: 1 2 3 6 5 4 Column-wise: 1 6 2 5 3 4							
3 3 1 1 1 2 2 2 3 3 3	Row-wise: 1 1 1 2 2 2 3 3 3 Column-wise: 1 2 3 1 2 3 1 2 3							
3.	<p>WAP that will take inputs of a 3 by 3 matrix into a 2D array. Now find the determinant of this matrix. <a href="http://www.mathsisfun.com/algebra/matrix-determinant.html">http://www.mathsisfun.com/algebra/matrix-determinant.html</a></p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>1 2 3 4 5 6 7 8 9</td><td>0</td></tr></table>	Sample input	Sample output	1 2 3 4 5 6 7 8 9	0	*		
Sample input	Sample output							
1 2 3 4 5 6 7 8 9	0							

4.	WAP that will take inputs of a n sized square matrix into a 2D array. Now show all the elements of its two diagonals. Reference: <a href="http://en.wikipedia.org/wiki/Main_diagonal">http://en.wikipedia.org/wiki/Main_diagonal</a>	*				
<table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>5 1 2 3 4 5 5 4 3 2 1 2 2 2 2 2 6 7 8 9 0 1 9 3 7 4</td><td>Major diagonal: 1 4 2 9 4 Minor diagonal: 5 2 2 7 1</td></tr></table>			Sample input	Sample output	5 1 2 3 4 5 5 4 3 2 1 2 2 2 2 2 6 7 8 9 0 1 9 3 7 4	Major diagonal: 1 4 2 9 4 Minor diagonal: 5 2 2 7 1
Sample input	Sample output					
5 1 2 3 4 5 5 4 3 2 1 2 2 2 2 2 6 7 8 9 0 1 9 3 7 4	Major diagonal: 1 4 2 9 4 Minor diagonal: 5 2 2 7 1					
5.	WAP that will take the size of an identity matrix from the user and generate the identity matrix into a 2D array. Finally display it. Reference: <a href="http://en.wikipedia.org/wiki/Identity_matrix">http://en.wikipedia.org/wiki/Identity_matrix</a>	*				
<table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>5</td><td>1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1</td></tr></table>			Sample input	Sample output	5	1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1
Sample input	Sample output					
5	1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1					
6.	WAP that will take inputs of two $m \times n$ sized matrix into two 2D array, suppose A and B. Now do $C = A + B$ . Finally display all the elements from matrix / 2D array C.	*				
<table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>2 3 1 2 3 2 3 4 1 1 1 2 2 2</td><td>2 3 4 4 5 6</td></tr></table>			Sample input	Sample output	2 3 1 2 3 2 3 4 1 1 1 2 2 2	2 3 4 4 5 6
Sample input	Sample output					
2 3 1 2 3 2 3 4 1 1 1 2 2 2	2 3 4 4 5 6					
7.	WAP that will take inputs of two $3 \times 3$ sized matrix into two 2D array, suppose A and B. Now do $C = A * B$ (multiplication). Finally display all the elements from matrix / 2D array C.	***				
<table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>1 2 3 4 5 6 7 8 9 2 2 2 2 2 2 1 1 1</td><td>9 9 9 24 24 24 39 39 39</td></tr></table>			Sample input	Sample output	1 2 3 4 5 6 7 8 9 2 2 2 2 2 2 1 1 1	9 9 9 24 24 24 39 39 39
Sample input	Sample output					
1 2 3 4 5 6 7 8 9 2 2 2 2 2 2 1 1 1	9 9 9 24 24 24 39 39 39					

8.	<p>WAP that will take inputs of <math>m \times n</math> sized matrix into a 2D array and find the maximum element with index location from that matrix.</p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>3 3 1 2 3 4 5 6 2 9 2</td><td>Max: 9 Location: [2][1]</td></tr><tr><td>2 3 9 8 7 3 4 5</td><td>Max: 9 Location: [0][0]</td></tr></table>	Sample input	Sample output	3 3 1 2 3 4 5 6 2 9 2	Max: 9 Location: [2][1]	2 3 9 8 7 3 4 5	Max: 9 Location: [0][0]	*																																																																										
Sample input	Sample output																																																																																	
3 3 1 2 3 4 5 6 2 9 2	Max: 9 Location: [2][1]																																																																																	
2 3 9 8 7 3 4 5	Max: 9 Location: [0][0]																																																																																	
9.	<p>WAP that will take <math>(n \times n)</math> integer inputs into a square matrix of dimension <math>n</math> (where <math>n</math> must be an odd number). Then calculate sum of the integers at first row, last row and two diagonals without overlap. Please see the sample input-output.</p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>5 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>2</td><td>3</td><td>4</td><td>1</td><td>6</td></tr><tr><td>3</td><td>4</td><td>9</td><td>6</td><td>7</td></tr><tr><td>4</td><td>2</td><td>6</td><td>7</td><td>8</td></tr><tr><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table></td><td>52</td></tr><tr><td>7 <table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table></td><td>23</td></tr></table>	Sample input	Sample output	5 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>2</td><td>3</td><td>4</td><td>1</td><td>6</td></tr><tr><td>3</td><td>4</td><td>9</td><td>6</td><td>7</td></tr><tr><td>4</td><td>2</td><td>6</td><td>7</td><td>8</td></tr><tr><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	1	2	3	4	5	2	3	4	1	6	3	4	9	6	7	4	2	6	7	8	5	4	3	2	1	52	7 <table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	23	**
Sample input	Sample output																																																																																	
5 <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>2</td><td>3</td><td>4</td><td>1</td><td>6</td></tr><tr><td>3</td><td>4</td><td>9</td><td>6</td><td>7</td></tr><tr><td>4</td><td>2</td><td>6</td><td>7</td><td>8</td></tr><tr><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	1	2	3	4	5	2	3	4	1	6	3	4	9	6	7	4	2	6	7	8	5	4	3	2	1	52																																																								
1	2	3	4	5																																																																														
2	3	4	1	6																																																																														
3	4	9	6	7																																																																														
4	2	6	7	8																																																																														
5	4	3	2	1																																																																														
7 <table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	23																																
1	1	1	1	1	1	1																																																																												
1	1	1	1	1	1	1																																																																												
1	1	1	1	1	1	1																																																																												
1	1	1	1	1	1	1																																																																												
1	1	1	1	1	1	1																																																																												
1	1	1	1	1	1	1																																																																												
1	1	1	1	1	1	1																																																																												

10.	<p>WAP that will take (n x n) integer inputs into a square matrix of dimension n (where n must be an odd number). Then calculate sum of the integers based on following position pattern (consider only the boxed position during the sum). Please see the input-output.</p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>5 <div><div>12345</div><div>23416</div><div>34967</div><div>42678</div><div>54321</div></div></td><td>71</td></tr><tr><td>7 <div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div></div></td><td>25</td></tr></table>	Sample input	Sample output	5 <div><div>12345</div><div>23416</div><div>34967</div><div>42678</div><div>54321</div></div>	71	7 <div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div></div>	25	**
Sample input	Sample output							
5 <div><div>12345</div><div>23416</div><div>34967</div><div>42678</div><div>54321</div></div>	71							
7 <div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div></div>	25							
11.	<p>WAP that will take (n x n) integer inputs into a square matrix of dimension n (where n must be an odd number). Then calculate sum of the integers based on following position pattern (consider only the boxed position during the sum). Please see the input-output.</p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>5 <div><div>12345</div><div>23416</div><div>34967</div><div>42678</div><div>54321</div></div></td><td>65</td></tr><tr><td>7 <div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div></div></td><td>33</td></tr></table>	Sample input	Sample output	5 <div><div>12345</div><div>23416</div><div>34967</div><div>42678</div><div>54321</div></div>	65	7 <div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div></div>	33	**
Sample input	Sample output							
5 <div><div>12345</div><div>23416</div><div>34967</div><div>42678</div><div>54321</div></div>	65							
7 <div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div><div>1111111</div></div>	33							

12.	<p>WAP that will take (m x n) integer inputs into a matrix of dimension m x n. Now reverse that matrix within itself and display it. Reversal means swap 1<sup>st</sup> column with the n<sup>th</sup> column, swap 2<sup>nd</sup> column with the (n-1)<sup>th</sup> column and so on...</p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>3 3 1 2 3 4 5 6 2 9 2</td><td>3 2 1 6 5 4 2 9 2</td></tr><tr><td>2 6 1 2 3 4 5 6 9 8 7 6 5 4</td><td>6 5 4 3 2 1 4 5 6 7 8 9</td></tr></table>	Sample input	Sample output	3 3 1 2 3 4 5 6 2 9 2	3 2 1 6 5 4 2 9 2	2 6 1 2 3 4 5 6 9 8 7 6 5 4	6 5 4 3 2 1 4 5 6 7 8 9	**
Sample input	Sample output							
3 3 1 2 3 4 5 6 2 9 2	3 2 1 6 5 4 2 9 2							
2 6 1 2 3 4 5 6 9 8 7 6 5 4	6 5 4 3 2 1 4 5 6 7 8 9							
13.	<p>WAP that will take (n x n) integer inputs into a square matrix of dimension n. Now determine whether the matrix is symmetric or not. Reference: <a href="http://en.wikipedia.org/wiki/Symmetric_matrix">http://en.wikipedia.org/wiki/Symmetric_matrix</a></p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>3 1 7 3 7 4 5 3 5 6</td><td>Yes</td></tr><tr><td>2 1 3 4 2</td><td>No</td></tr></table>	Sample input	Sample output	3 1 7 3 7 4 5 3 5 6	Yes	2 1 3 4 2	No	**
Sample input	Sample output							
3 1 7 3 7 4 5 3 5 6	Yes							
2 1 3 4 2	No							
14.	<p>WAP that will take (m x n) positive integer inputs into a matrix of dimension m x n. Now replace all the duplicate integers by -1 in that matrix. Finally display it.</p> <table><tr><th>Sample input</th><th>Sample output</th></tr><tr><td>3 3 1 7 3 7 4 5 3 5 6</td><td>1 7 3 -1 4 5 -1 -1 6</td></tr><tr><td>2 6 2 2 2 2 2 2 6 5 4 3 2 1</td><td>2 -1 -1 -1 -1 -1 6 5 4 3 -1 1</td></tr></table>	Sample input	Sample output	3 3 1 7 3 7 4 5 3 5 6	1 7 3 -1 4 5 -1 -1 6	2 6 2 2 2 2 2 2 6 5 4 3 2 1	2 -1 -1 -1 -1 -1 6 5 4 3 -1 1	***
Sample input	Sample output							
3 3 1 7 3 7 4 5 3 5 6	1 7 3 -1 4 5 -1 -1 6							
2 6 2 2 2 2 2 2 6 5 4 3 2 1	2 -1 -1 -1 -1 -1 6 5 4 3 -1 1							

15.

WAP that will take (m x n) integer inputs into a matrix of dimension m x n. Now just simply add all the integers in that matrix and show the result.

\*

Sample input	Sample output
3 3 1 7 3 7 4 5 3 5 6	41
2 6 2 2 2 2 2 2 6 5 4 3 2 1	33