

**Islamic University of Technology (IUT)**  
Organization of Islamic Cooperation (OIC)  
Department of Electrical and Electronic Engineering (EEE)

Time: 3.00 hrs

**Exercise - 01:**

Given a matrix of size (m,n) –

Return a new matrix encircling the old matrix by 100.

8	1	6
3	5	7
4	9	2

100	100	100	100	100
100	8	1	6	100
100	3	5	7	100
100	4	9	2	100
100	100	100	100	100

## Exercise - 02:

Given an array of integers, sort the numbers based on their score\*.

\*The score is the summation of all digits in that number.

For example, for  $n = 38$ , its score is  $= 3+8 = 11$ .

### Test Case – 01:

- Input: [99, 1000, 237, 6 , 72001]
- Output: [ 1000, 6, 72001, 237, 99]

### Test Case – 02:

- Input: [5, 7 ,9, 11, 14 ,29]
  - Output: [ 11, 5, 14, 7, 9, 29 ]
- 
- If two numbers have the same score, place them in accordance with their original index in the given array.
  - Hence, in test case – 02,  
5 and 14 both have same weight but 5 appeared prior to 14 in the original array. So, in the output, it is placed before 14.

### Exercise - 03:

In a chessboard of size (8,8) – a knight is placed in a given location [x, y].

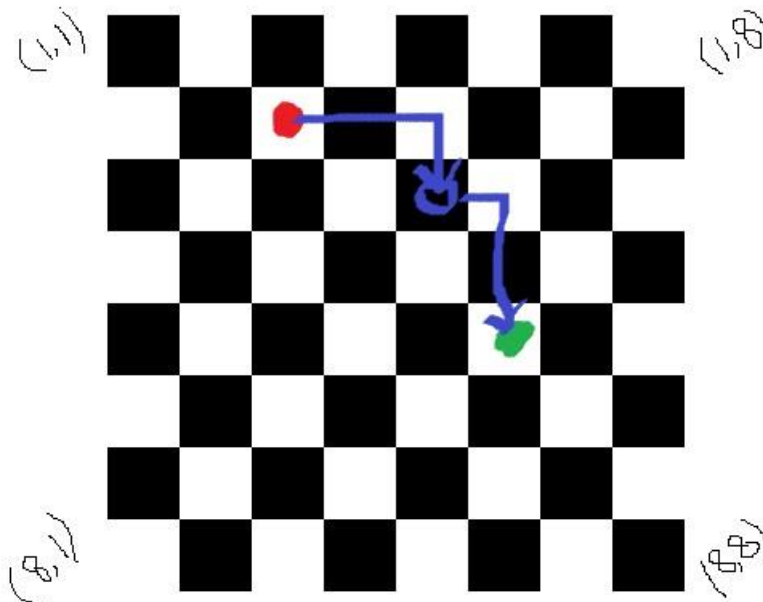
A target location [p, q] is also provided.

The knight wants to reach that target with **minimum** number of moves.

Find the minimum no of moves the knight can take to reach the location [p,q] from the location [x,y].

#### Test Case - 01:

- Input: loc= [2,3], target = [5,6]
- Output: 2



- The red marked position is the initial location [2,3]
- The green marked position is the target [5,6]
- A knight has 8 possible moves. So, there are different ways the knight can reach the destination. You have to find the optimal way.

#### **Exercise - 04:**

Given an array of integers, find the length of the longest decreasing sub-sequence.

For example, a = [2 ,4 ,3, 1, 5, 7, 2 ,11, 12]

Here, [4, 3, 1] and [ 7, 2] are two decreasing sub-sequence.

So, the length of the longest decreasing sub-sequence is 3.

#### **Test Case – 02:**

- Input: [ 1, 2, 3, 5, 7, 5, 4, 0 ]
- Output: 4

#### **Test Case – 03:**

- Input: [ 1 ,2, 4, 6, 13, 32, 56, 89, 1532]
- Output: 1

### **Exercise - 05:**

In American style, date is written in the format Month – Day – Year.

You're given a date as string (character array).

Convert it to British style. Your output should also be a character array.

#### **Test Case – 01:**

- Input: 'Oct 31, 2020'
- Output: '31 Oct, 2020'

#### **Test Case – 02:**

- Input: 'Jan 1, 19'
- Output: '1 Jan, 19'

#### **Test Case – 03:**

- Input: 'Feb 08, 06'
- Output: '08 Feb, 06'

### Exercise - 06:

Given an matrix of size (m,n) –

Find the numbers in the matrix that are minimum in their respective row and column.

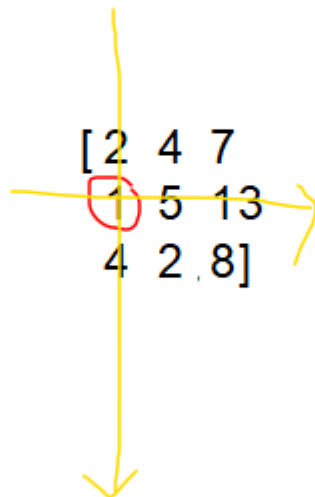
For example,

Mat = [ 2 4 7

1 5 13

4 2 8]

Output: [1, 2]



Here, as you can see, 1 is minimum element in 2<sup>nd</sup> row and 1<sup>st</sup> column. [ index of 1 = (2,1) ]

Similarly, 2 is minimum in 3<sup>rd</sup> row and 2<sup>nd</sup> column. [ index of 2 = (3,2) ]

### Exercise - 07:

Write a function that will take two inputs: an array of integers and a string ('left' or 'right') place a 0 after or before each one of the inputs.

For example, a = [ 2, 4, 5] and s = 'left'

Output: [0,2,0,4,0,5]

### Test Case – 02:

- Input: a = [5, 7, 9, 13] and s = 'right'
- Output: [5,0,7,0,9,0,13,0]

- The string will always be 'left' or 'right'. You don't have to worry about other types.

## Exercise - 08:

Given a cell array of words, find the longest 3 words. Return them as a cell array.

### Test Case - 01:

- Input: { 'max' , 'payne', 'call' , 'of', 'duty' }
- Output: {'payne', 'call', 'duty'}

### Test Case - 02:

- Input: { 'logan', 'wolverine', 'storm', 'magneto', 'charles', 'sophiee' }
- Output: { 'wolverine', 'magneto', 'charles' }

- The output array should have the elements in the same sequence as the input.
- Here, 'magneto', 'charles' and 'sophiee', all have length 7. But since you've only two spots left, return the first two.

### Test Case - 03:

- Input: { 'aa', 'bbb', 'c', 'dddd', 'ee' }
- Output: { 'aa', 'bbb', 'dddd' }

- The output is not to be sorted.



### Exercise - 09:

A 3D matrix of size (4,2,n) is given.  $n \geq 2$ .

First two dimensions represent coordinates in the xy plane.

Find the area and perimeter of the quadrilaterals.

**Test Case – 01:** For a given matrix  $z$  of size (4,2,4), output should be of size (4,2). Here the 1<sup>st</sup> column represents the area of each quadrilateral while the 2<sup>nd</sup> column represents the perimeter.

$z(:, :, 1) =$

0	-2
4	3
3	7
-4	-4

$z(:, :, 2) =$

2	9
3	3
9	10
3	4

$z(:, :, 3) =$

-6	-3
-5	7
2	2
-1	1

$z(:, :, 4) =$

5	8
-5	10
0	1
4	-8

out =

25.5000	28.0368
3.5000	28.8866
34.0000	28.2176
76.5000	46.3737

## Exercise - 10:

A list of assignments is given to the students.

- Each of the assignment carries equal marks.
- Different assignments take different amounts of time to complete.
- But for each day of delay, a penalty is added.
- Only one assignment can be done at a time.

For example,

- Assignment = [ a1, a2, a3, a4, a5, a6]
- Time Req. = [ 2, 4, 3, 8, 1, 10]
- Penalty = [ 10, 4, 1, 2, 5, 2]

Now, say if he starts with the 1st assignment - then a penalty will be added to all the other assignments for two days since it takes 2 days to complete the 1st assignment.

Then say he starts the 2nd assignment which takes 4 days to complete. So penalty will be added for the other assignments that he hasn't started yet (not the 1st one--since it's been finished)

In which order should he finish his assignments so that he has to suffer the minimum penalty.

For the example above, the order should be= [1,5,2,3,4,6] .

[say, I start 1st assign. today -- now it'll take me 2 days to complete. once I start with one assign, I've to finish it first - then jump to the next assign.

Now for these 2 days, I couldn't 'start' the other assign. so I've to suffer given penalty for each assign. each day. that is =  $4*2 + 1*2 + 2*2 + 5*2 + 2*2$  .

After 2 days, say I start the last assign. which will take 10 days to finish. So, for each of these 10 days - the penalty will keep adding for those assigns. that I've not yet started.

this way continue.

At the end, a particular amount of penalty will be accumulated. That total penalty has to be minimum.

there'll be many combinations - among all those, which one will give minimum penalty ]

- Input: time= [ 2, 4, 3, 8, 1, 10] , penalty = [ 10, 4, 1, 2, 5, 2]
- Output: [1,5,2,3,4,6]