

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

Time: 3.00 hrs

Instructions:

Read the instructions carefully. Don't deviate from the instructions since we'll be using auto-checking process.

- You need to write a function for each of the questions. [not script]
 - The function name should be of the format exc(question no.) e.g. 'exc01'
 - The name of the m-files should be same as the function names.
 - You have to return all your m-files in a zip file named as your student ID.
-
- Follow the input and output data-type exactly the way it is shown in the test cases.

For example, if output is shown as True, then your function needs to return logical True, not string 'true'.

More importantly, printing the output using disp or fprintf will not work. Because these functions only display the output, don't return any value to the function.

To return a value to your function, you must assign it to your function's output variable.

So, even if your code is right, without properly returning the value to your function – the checker will mark it as wrong.

- Don't hardcode your program. In addition to the test cases shown in the questions, we'll keep some hidden test cases for examining your result.
- So, your code will fail if you solve the problems manually or use lookup table to return value to your function.

Exercise - 01:

Given a matrix of size (m,n) – extract the 2nd layer in a cyclic way.

For example, for the matrix in the figure, the output should be –

Out = [7,8,9,14,19,18,17,12]

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

Test Case – 02:

- Input: x= magic(5)
 - Output: [5, 7, 14, 20, 21, 19, 12, 6]
-
- Output has to be a row vector
 - m,n >=3
 - Careful, there shouldn't be any repetitions.

🚦 Function_name: exc01

Exercise - 02:

Three row vector a, b, c is given. They are of same size.

- I. Create a table from them [so your table should have 3 columns].
- II. Add a new column named 'max' to your table that contains the maximum of the three.
- III. Add a new column named 'min' to your table that contains the minimum of the three.
- IV. Sort the rows based on the 3rd column.

Test Case:

- Input: a= [2,3,4], b= [6,2,4], c= [9,14,2]
- Output:

After Step - 03

t =

3×5 table

| a | b | c | max | min |
|---|---|----|-----|-----|
| — | — | — | — | — |
| 2 | 6 | 9 | 9 | 2 |
| 3 | 2 | 14 | 14 | 2 |
| 4 | 4 | 2 | 4 | 2 |

After Step - 04

3×5 table

| a | b | c | max | min |
|---|---|----|-----|-----|
| — | — | — | — | — |
| 4 | 4 | 2 | 4 | 2 |
| 2 | 6 | 9 | 9 | 2 |
| 3 | 2 | 14 | 14 | 2 |

- The final table should be returned as output.

 Function_name: exc02

Exercise - 03:

Given an integer n, create a 'X' shape output using asterisk (*).

For example, for n=5

Output should look like this -

5×5 char array

```
'*'  '*'  
'*'*'  
' '*'  
'*'*'  
'*'  '*'
```

- The empty places are not space but blank/null character (ascii value = 0)

🚦 Function_name: exc03

Exercise - 04:

Given a string which is a mail address, e.g. 'abcd@iut-dhaka.edu'

find the following –

- I. The user name ['abcd']
- II. Change the .edu part to .com ['abcd@iut-dhaka.com']
- III. Find the institution name ['iut']

Test Case – 02:

- Input: 'sssss@uct-uk.edu'
- Output:
 - i. 'sssss'
 - ii. 'sssss@uct-uk.com'
 - iii. 'uct'

- The email will always be in the aforementioned format.
- Input will be a character array; so should be the output.

 Function_name: exc04

Exercise - 05:

Write a function that will convert uppercase letters into lowercase from a character array.

- Do not use any built-in function in your code e.g. upper, lower, regexp, isstrprop, isupper, islower etc.

Test Case - 01:

- Input: 'AbcD567Zoiu'
- Output: 'abcd567zoiu'

Test Case - 02:

- Input: 'X-Men'
- Output: 'x-men'

- If you use any of those built-in functions, the checker will automatically grade it 0.

🚦 Function_name: exc05

Exercise - 06:

Say you want to withdraw some money from the ATM booth in IUT and the machine provides 1000, 500 and 100 taka notes. Now, the machine always wants to return minimum no. of notes i.e. it prefers to give one 1000 tk notes compared to giving ten 100tk notes.

To withdraw x (input) amount of money, what is the **minimum** number of notes the atm can provide?

For example, if you want to withdraw 2600 taka, then the answer should be 4 [2 – 1000tk, 1 – 500tk, 1-100tk].

You can also get 2600 taka by other ways but that will not be optimal way.

Test Case:

- Input: 4900
- Output: 9

 **Function_name:** exc06

Exercise - 07:

Write a function that will take a 2D matrix as input, and it will return the output as a row matrix with all the elements associated in a spiral pattern.

➤ Input:

[1 2 3

4 5 6

7 8 9]

➤ Output: [1,2,3,6,9,8,7,4,5]

- The matrix is of size (m,n)

Test Case – 02:

▪ Input: [16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1]

▪ Output: [16,2,3,13,8,12,1,15,14,4,9,5,11,10,6,7]

Hint: `rot90()` rotates your matrix by 90 degree.

🚦 Function_name: `exc07`

Exercise - 08:

An array is given (even no. of inputs). Encode it in the following way –

Input: [3, 2, 1, 5, 4, 1]

Output: [2,2,2, 5, 1,1,1,1]

Explanation: Look at the input array elements pair-wise.

- 1st pair is (3, 2) == three, 2's.
- 2nd pair is (1, 5) == one, 5.
- 3rd pair is (4,1) == four, 1's.

Test Case – 02:

- Input: [2, 9]
- Output: [9, 9]

Test Case – 03:

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

 Function_name: exc08

Exercise - 09:

A vampire number(v) is a number with even number of digits that has a factorization $v = i * j$.

Here, i and j have to satisfy the following conditions –

- Both of them have half as many digits as the original number v .
- Maximum one of them can be divisible by 10.
- The digits in v consists of the digits of x and y .

i and j are called the fangs of the number v .

For example,

- $1260 = 21 * 60$
 - ✓ $v=1260$ has 4 digits while 21 and 60 have 2 digits (half).
 - ✓ Only one of them is divisible by 10.
 - ✓ Digits in 21 and 60 make 2160.
- $1260 = 28 * 45$
 - ✓ 1st condition is satisfied.
 - ✓ 2nd condition is also satisfied.
 - ✓ 3rd condition fails.

Hence, [21,60] are the fangs of 1260. While [28,45] are not.

Given a vampire number, find the fangs of that number.

In case of multiple fangs, return only one.

Test Case – 02:

- Input: 150300
- Output: [300, 501]

 Function_name: exc09

Exercise - 10:

The following MATLAB code produces a **multiplication of two Black and white image**. See the illustration-1. Here you need to **load the image at first**. As you want to multiply the two images; you need to convert them into a matrix of numerical values, **rgb2gray** will perform that task. Now as the images only have either black or white colors, so you can use **im2bw** for making this **black color = 1** and **white color= 0**. Your **im2bw** is a function which will return you **logical parameters**; as you can't multiply logical parameters, you need to convert them into **double**. Then you are ready to multiply. Right after multiplication, in order to show the image, you need to **convert your double variable into logical**. Now the given code is not arranged in sequence, hence re-arrange the sequence.

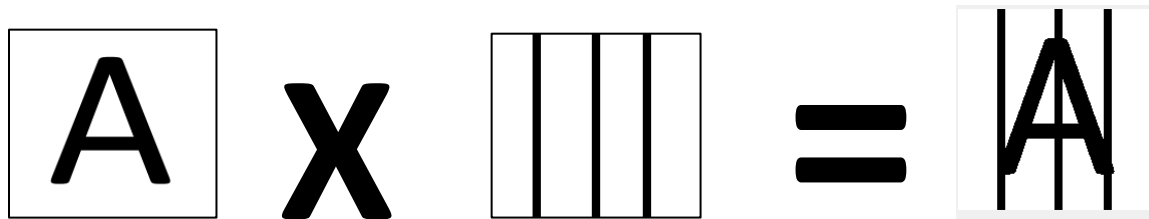


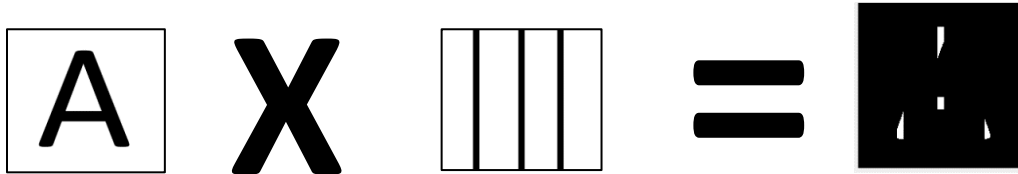
Illustration 1

```
1. imgx=imread('A.PNG');
2. imgi=im2bw(img1,graythresh(imgd));
3. imgd=rgb2gray(imgx);
4. imgy=imread('vertical.PNG')
5. imgf=rgb2gray(imgy);
6. imgi=im2bw(img1,graythresh(imgd));
7. imm = double((imgi));
8. imgl=im2bw(img2,graythresh(imgf));
9. test=logical(test)
10.      test=imn.*imm;
11.      imn = double((imgl));
12.      figure,imshow(test)
```

correct sequence:

| | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|

Now write a code to make the following modification of the previous code.



 Function_name: exc10