

Instructions for students:

1. Complete the following problem using concepts of **RECURSION**
2. You may use any language to complete the tasks.
3. You need to submit one single file containing all the methods/functions. You will get one week to complete your lab. **NO LATE SUBMISSIONS WILL BE CONSIDERED.**
4. The submission format **MUST** be maintained. You need to copy paste all your codes in ONE SINGLE .ipynb file and upload that. If format is not maintained, whole lab submission will be canceled.
5. 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.
6. If you are using PYTHON, make sure your code has the methods invoked and proper printing statements according to the tasks.
7. **NO USE OF LOOP OR ANY OTHER BUILT IN FUNCTION UNTIL EXPLICITLY MENTIONED**
8. You may use more than one method to solve these problems.
9. You may use default parameters if required.

Assignment Tasks: [CO4]

1. [Very Easy] [5 Marks each]

a) Implement a recursive algorithm that takes an array and adds all the elements of the array.

b) The total number of combinations of size r from a set of size n [where r is less than or equal to n], we use the combination nCr where the formula is:

$$C(n,r) = \frac{n!}{r!(n-r)!}$$

Given n and r [$r \leq n$], write a recursive function to find out nCr .

Apart from finding the factorial **recursively**, there is another recursive formula:

$$C(n,r) = C(n-1, r-1) + C(n-1, r)$$

Where if $n == 0$ or $n == r$, $C(n,r) = 1$

We can solve the task using any formula you want.

c) Count the number of digits in a given number.

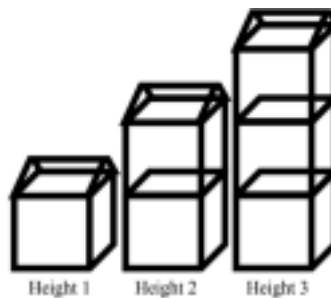
d) Given a number, check whether the number is prime or not.

2. [Easy] [5 Marks each]

- a) Implement a recursive algorithm that takes a decimal number n and converts n to its corresponding (you may return as a string) hexadecimal number.
- b) Write a recursive function that takes an array and a **singly linked list head** as parameters. The method compares between the array and the Linkedlist. If all the elements are identical in both then it will return True otherwise, False. The length of Array and LinkedList will be the same.
- c) Write a Recursive function that takes a **singly linked list head** as parameter and prints all the vowels in the given string in the **same** order as they appear and then all the consonants in the **reverse** order. You can assume that the string will only consist of alphabets (a-z, A-Z), i.e. no space, special character, or numeric character will be in the string.

3. [Medium] [8 Marks Each]

- a) Suppose, you have been given a non-negative integer which is the height of a 'house of cards'. To build such a house you at-least require 8 cards. To increase the level (or height) of that house, you would require four sides and a base for each level. Therefore, for the top level, you would require 8 cards and for each of the rest of the levels below you would require 5 extra cards. If you were asked to build level one only, you would require just 8 cards. Of course, the input can be zero; in that case, you do not build a house at all. Complete the **recursive method** below to calculate the number of cards required to build a 'house of cards' of specific height given by the parameter.



```
public int hocBuilder (int height){  
    // TO DO  
}
```

OR

```
def hocBuilder(height):  
    #TO DO
```

- b) Given an array of integers, write a recursive function to find out the maximum number in that array.

4. [Hard] [10 + 10 = 20 Marks]

a. Print the following pattern for the given input (**you must use recursion**):

Sample Input: 5

Sample Output:

1				
1	2			
1	2	3		
1	2	3	4	
1	2	3	4	5

b. Print the following pattern for the given input (**you must use recursion**):

Hint:

i. The first element of a row is 2 more than the first element of the previous row. The first element of row 1 is 1.

ii. Each element of row 1 has difference of 1

Each element of row 2 has difference of 2

Each element of row 3 has difference of 4

Each element of row 4 has difference of 8

And so on and so forth

Test Case 1:

Sample Input: 5

Sample Output:

1	2	3	4	5
3	5	7	9	
5	9	13		
7	15			
9				

Test Case 2:

Sample Input: 6

Sample Output:

1	2	3	4	5	6
3	5	7	9	11	
5	9	13	17		
7	15	23			
9	25				
11					

5. [Very Hard] [12 Marks]

Complete the recursive function `flattenList` which will take a nested python list and an empty list as parameters. The function should convert the nested list into a new flat list sequentially. Check the input-output to understand the sequence pattern.

```
def flattenList(given_list, output_list):  
    # To Do  
  
given_list = [1, [2, [3, [4], 5], 6], 7, 8, [9, [[10, 11], 12], 13], 14, [15, [16, [17]]]]  
output_list = flattenList(given_list, []) # Initial empty list is sent for update  
print(output_list)  
#output_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
```

Your code must work for any depth of the nested array. You can not use any loop in the function. You **can use** built-in functions like `append`, `type`, etc.