BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Midterm | **Semester**: Fall 2023
Duration: 80 Minutes | **Full Marks**: 30
No. of Questions: 3 | **No. of Pages**: 2

CSE 220: Data Structures

Name: (Please write in CAPITAL LETTERS)	ID:	Section:

A

✔ Use the back **part** of the answer script for rough work. **No washroom breaks.**
✔ At the end of the exam, put the question **paper** inside the answer script and **return both**.
✔ **Understanding questions is part of exam.**

Question 1: CO1, CO5 [2 + 8 Points]

I. Suppose you are given a multi-dimensional array with dimension 5x5x3. What is the multidimensional index for the linear index 65?

II. **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 rows and columns 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]]

Question 2: CO3 [2 + 8 Points]

I. **Write** down two disadvantages of Linked List over Array.

II. You are given two linked lists of the same even length. Your task is to **complete** a method **pairwiseEqual()** that takes two singly linear linked list heads as arguments, checks if the linked lists are equal pairwise and returns True/False.
Two linked lists will be equal pairwise if the node values of every pair in Linked List 1 are equal to the node values of corresponding pair in Linked List 2 irrespective of their sequence **[i.e. the sequence does not matter].**
[DO NOT USE OTHER DATA STRUCTURE OTHER THAN GIVEN LINKED LISTS]

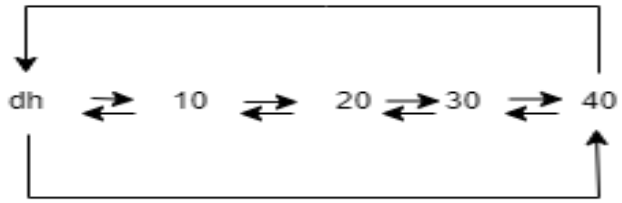
Sample Input	Returned Value	Explanation
head1 =10-->15-->34-->41 head2 = 15-->10-->34-->41	True	Linked List 1 Pairs: (10,15) ,(34,41) ; Linked List 2 Pairs: (15,10) , (34,41)
head1 =10-->15-->34-->42 head2 = 15-->10-->34-->41	False	Linked List 1 Pairs: (10,15) ,(34,42) ; Linked List 2 Pairs: (15,10) , (34,41)

Python Notation	Java Notation

def pairWiseEqual(h1, h2):
    # To Do
	public boolean pairWiseEqual(Node h1, Node h2) {
	    // To Do
	}

Question 3: CO1, CO5 [5 + 5 Points]

I. You are given a dummy headed doubly circular linked list and a block of code. The list is like below:




```
for i in range(5):
    for j in range(i):
        n1 = dh.next
        n2 = n1.next
        n3 = dh.prev
        dh.next = n2
        n2.prev = dh
        n3.next = n1
        n1.next = dh
        n1.prev = n3
        dh.prev = n1
```

Draw the resulting list (The list you will find after the nested loop) for each value **i** in your answer script. Mention the value of **i** and the resulting list, no need to show the intermediate states.

II. Consider that a **MidStack** class has been created containing the push(element), pop(), peek() and isEmpty() functions. **No need to implement MidStack class.** The MidStack class implements a singly linked list-based Stack hence overflow is not possible. The pop() and peek() functions return None in case of the underflow.

Complete the function **conditional_reverse** which will take an object of MidStack that contains some integer values. The function returns a new stack which will contain the values in reverse order from given stack with the exception that if consecutive numbers are the same, it picks only one from them. **You cannot use any other data structure except MidStack.**
Remember that a stack has no other functions than push, pop, peek, and isEmpty

Python Notation	Java Notation

def conditional_reverse(stack):
    # To Do
	public MidStack conditional_reverse(MidStack stack) {
	// To Do
	}

Sample Input Stack (Right most is the top)	Returned Stack (Right most is the top)	Explanation
Stack: 10, 10, 20, 20, 30, 10, 50 Top = 50	Stack: 50, 10, 30, 20, 10 Top = 10	Consecutive 20 and 10 are not present in the output reversed stack