

A

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Mid-Term
Duration: 80 Minutes
Number of Questions: 3

CSE220: Data Structures

Semester: Spring 2024
Full Marks: 30
No. of Pages: 3

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

- ✓ **Answer all 3 questions. No washroom breaks.**
- ✓ **At the end of the exam, put the question paper inside the answer script and return both.**

Question 1: CO1 [2 + 8 Points]

- I. Suppose you are given a multi-dimensional array with dimensions 4x5x4. What is the linear index of the multidimensional index [2][1][0]?
- II. Suppose you're working as a cryptographer for a secret intelligence agency. You are given an encrypted matrix as an input. You have to decrypt it after some processing of this matrix.

To decrypt this message efficiently, you have a task to construct a method called **sum_diff(matrix)** which takes an encrypted matrix as input and returns a decrypted linear array. The process of finding the decrypted linear array is given below:

You have to find out the column-wise summations for each column and store the difference of subsequent column-wise summations in a new linear array.

Sample Input	Sample output	Explanation																			
<table><tr><td>1</td><td>3</td><td>1</td></tr><tr><td>6</td><td>4</td><td>2</td></tr><tr><td>5</td><td>1</td><td>7</td></tr><tr><td>9</td><td>3</td><td>3</td></tr><tr><td>8</td><td>5</td><td>4</td></tr></table>	1	3	1	6	4	2	5	1	7	9	3	3	8	5	4	<table><tr><td>-13</td><td>1</td></tr></table>	-13	1	<p>Sum of 0th column = 29 Sum of 1st column = 16 Sum of 2nd column = 17</p> <p>Therefore, the size of the resulting array is 2 and the array is:</p> <table><tr><td>16-29 = -13</td><td>17-16 = 1</td></tr></table>	16-29 = -13	17-16 = 1
1	3	1																			
6	4	2																			
5	1	7																			
9	3	3																			
8	5	4																			
-13	1																				
16-29 = -13	17-16 = 1																				

Python Notation	Java Notation
<pre>def sum_diff(matrix): # To Do</pre>	<pre>public int[] sum_diff(matrix) { // To Do }</pre>

Question 2: CO5 [10 Points]

You are given a non-dummy headed singly linear linked list containing positive integers. Your task is to complete the given method **IsSumPossible()** which takes the head of the linked list and an integer, n as input. If the integer n can be obtained by summing any two of the elements of the linked list, return True, else return False.

Hint: There might be more than one such pair possible, you don't have to check all such pairs.

[DO NOT USE OTHER DATA STRUCTURE OTHER THAN GIVEN LINKED LIST]

Sample Input	Returned Value	Explanation
list = 1→2→3→4→5 n = 7	True	Sum of the elements (3,4) or (2,5) equals 7
list = 1→2→4→5→6 n = 4	False	There are no two elements that make the sum 4. Note that though there is an element 4 itself in the list, the output will be False.
list = 5 n = 5	False	There is only one element so, the sum of two elements cannot be 5

Python Notation	Java Notation
<pre>def IsSumPossible(head, n): # To Do</pre>	<pre>public boolean IsSumPossible(Node head, int n) { // To Do }</pre>

Question 3: CO3 [2+8 Points]

- I. Suppose that an intermixed sequence of stack push and pop operations are performed. The pushes push the integers 0 through 9 in order; the pops print out the return value. Which of the following sequences could not occur for push and pop operation? Write all the correct answers.
- A. 1 2 3 4 5 6 9 8 7 0
 - B. 0 4 6 5 3 8 1 7 2 9
 - C. 1 4 7 9 8 6 5 3 0 2
 - D. 2 1 4 3 6 5 8 7 9 0
- II. You are given a **Stack** of people. They should be paired off as man with woman, woman with man. If a male and a female are consecutive in the stack, they get paired immediately. If there are consecutive females found without any subsequent males, then the spare females should be saved in a **Queue** for future pairing. If there are consecutive males found without any subsequent females, we try to pair a male with the first spare female [if available] from the **Queue**. If no spare female is available, then the male is discarded. The same logic applies to consecutive females without any subsequent males.

The stack class and the Queue class are implemented for you. **The public instance methods in the Stack class are push(), pop(), peek(), and isEmpty(). The public instance methods in the Queue class are enq(), deq(), peek(), and isEmpty().** In both classes, isEmpty() returns a boolean value, peek() and pop() returns None for underflow. All other members of both classes are private. You can only use the public instance methods. No need to consider the overflow exception for both Stack and Queue classes.

Your task is to complete the given method **dance_pair()** that takes a stack and prints every male-female pair.

Every male and female are written in this way - M_id or F_id. A method called **id_gender_extractor(s)** is implemented for you that returns the id and gender from a string,s.

You do not need to write the methods/classes that are implemented for you. You cannot use any data structure other than stack and queue.

Sample Input	Output	Explanation										
<div>st =</div> <div>Top →</div> <table><tr><td>M_10</td></tr><tr><td>F_20</td></tr><tr><td>F_4</td></tr><tr><td>F_5</td></tr><tr><td>M_3</td></tr><tr><td>M_19</td></tr><tr><td>M_1</td></tr><tr><td>M_7</td></tr><tr><td>F_9</td></tr><tr><td>F_18</td></tr></table>	M_10	F_20	F_4	F_5	M_3	M_19	M_1	M_7	F_9	F_18	10 and 20 are paired together 5 and 3 are paired together 4 and 19 are paired together 7 and 9 are paired together	M_10 and F_20, F_5 and M_3, M_7 and F_9 are sequential in the stack. They are paired up. From F_4, there are consecutive females. So, F_4 is stored in the queue for future pairing. From M_19, there are consecutive males. M_19 is paired with spare F_4. M_1 is discarded since there is no spare female with whom M_1 can be paired.
M_10												
F_20												
F_4												
F_5												
M_3												
M_19												
M_1												
M_7												
F_9												
F_18												

Given:

<pre>def id_gender_extractor(s): if s != None: return s[0],s[2:] #gender, id else: return None, None</pre>	<p>Driver Code:</p> <pre>st = Stack() st.push('F_18') st.push('F_9') st.push('M_7') st.push('M_1') st.push('M_19') st.push('M_3') st.push('F_5') st.push('F_4') st.push('F_20') st.push('M_10')</pre> <p>dance_pair(st)</p>
--	--

Python Notation	Java Notation
<pre>def dance_pair(st): # To Do</pre>	<pre>public void dance_pair(Stack st) { # To Do }</pre>