

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Semester Final
Duration: 1 Hours 45 Minutes

Semester : Summer 2022
Full Marks: 45

CSE 220: Data Structures

Answer all of the following questions.
Figures in the right margin indicate marks.

Name: <u>HASHRUB SAFIN SHAKIB</u>	ID: <u>20241037</u>	Section: <u>14</u>
-----------------------------------	---------------------	--------------------

1.
CO4

Adam wants to calculate the harmonic sum of any positive integer by using **recursion**. But he does not know how to **write** recursive code. He just knows that harmonic sum is the sum of reciprocals of the positive integers.

That means, the basic formula of harmonic sum of

$$n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

So, you have a task to help Adam by writing a **recursive program** in python/java to calculate the **harmonic sum** of any positive integer, n.

python:

```
def harmonic_sum(n):  
    #write code here
```

java:

```
public float harmonic_sum(int n){  
    //write code here  
}
```

Sample Input	Sample Output
harmonic_sum(4)	2.08333
harmonic_sum(7)	2.59286

$$\frac{1}{4} + \frac{1}{3} + \frac{1}{2} + \frac{1}{1} + 0$$

$$\frac{1}{4} + \frac{1}{3} + \frac{1}{2} + \frac{1}{1} = \frac{13}{12}$$

$$\frac{1}{4}$$

$$\frac{1}{2-1}$$

$$\frac{1}{1-1}$$

$$\frac{1}{2-1}$$

$$\frac{1}{1-1}$$

$$\frac{1}{2} + \frac{1}{2-1}$$

$$\frac{1}{2}$$

$$\frac{1}{2-1} = \frac{1}{1} \text{ 1 of 4}$$

2.

In a sequence, an **alphanumeric character** means to enqueue, a _(underscore) means to dequeue and a :(colon) means to peek. Now consider the following sequence:

A 9 : : e s X _ : _ _ : D W u _ : e K : _ _ : L _ _ :

A queue has been implemented by a circular array of capacity = 8 and front = 2.

CO1

- a. For the above-mentioned queue, **complete** the enqueue and dequeue method.

3+2

python:

```
def enqueue(cir_arr, front=2, size, element):
```

Write your code

```
def dequeue(cir_arr, front=2, size):
```

Write your code

Java:

```
public void enqueue(char[] cir_arr, int front, int size, char element){
```

// Write your code

```
}
```

```
public char dequeue(char[] cir_arr, int front, int size){
```

// Write your code

```
}
```

CO3

- b. **Simulate** the operations of the given sequence in the above-mentioned queue. You need to show the array change and also the enqueued, dequeued or peeked elements at each step

5

Name

Cor

Se

3.
CO2

- a. Suppose you are making a hash table for the students of the CSE department at BRAC University. The student's ID consists of eight digits and these are taken as the key for your hash function. You have decided to create an array of size 2999 for your hash table and to resolve collisions you will use linear probing. Consider the hash function to be the summation of digits of the student ID mod 2999.

For example: Student ID: 20125105, hash value will be $(2+0+1+2+5+1+0+5)\%2999 = 16$

Create a hash function that will generate the hash value for any given student id.

- b. Given the array S containing some names. We want to perform Key Index Searching but as the values are Strings we first have to convert them into integers. Hence we have designed a conversion function. Let A = 1, B = 2, ... Z = 26. The conversion function **conFun(element)** is defined by (summation of the integers associated with each character) % 10. If the element is ABC, **conFun(ABC)** = $(1+2+3)\%10$, which is 6.

- I. Your job is to generate an array named EL which will have the same length as S. Then for every element e in S, generate an integer and store it in EL inside the same index number containing e. The first one is done for you.

S =

ABC	EDC	JKL	LKM	CAB	ACD	MLK	EGH	CED	IJK	BAC
✓	✓	✓	✓	✓	✓	✓	✓	✓		

EL =

6	2	3	6	6	8	6	0	2	0	6
---	---	---	---	---	---	---	---	---	---	---

- II Show the auxiliary array from the result you have obtained from part I of the question.

A → 1
B → 2
C → 3
D → 4
E → 5
F → 6
G → 7
H → 8
I → 9
J → 10
K → 11
L → 12
M → 13
N → 14
O → 15
P → 16
Q → 17
R → 18
S → 19
T → 20
U → 21
V → 22
W → 23
X → 24
Y → 25
Z → 26

12/10

4
10 | 12
10
2

3
10 | 33
30
3

10 | 36
30
6

2
10 | 6

10 | 20
20
0

10 | 30
30
0

10 | 12
10
2

4.
CO2

- a. i. Draw a Binary Search Tree (BST) from the following sequence: 3
~~50, 20, 55, 60, 30, 25, 27, 109, 10, 45, 70~~
 ii. Show the output of the drawn trees using Inorder and Postorder traversal. 2
 iii. Remove 109 from the tree generated in a and draw the resultant tree 1
 iv. Remove 20 from the resultant tree generated in c using Successor and draw the resultant tree 2
 v. Remove root from the resultant tree generated in d using Successor and draw the resultant tree 2
- b. Construct a directed graph from the following Adjacency Matrix 3

	A	B	C	D	E	F
A	0	1	0	1	0	0
B	1	0	1 ✓	0	1 ✓	0
C	0	0	1 ✓	0	0	0
D	1 ✓	0	1 ✓	0	1 ✓	0
E	0	1 ✓	1 ✓	1 ✓	0	0
F	1 ✓	1 ✓	1 ✓	1 ✓	1 ✓	0

- c. Construct the Adjacency List from the resultant graph from b. 2

Ans