

Exam: 9:00 am – 12:00 pm, Uploading paper to LMS 12:00pm to 12:30pm

Course Code: CS 210/218	Course Name: Data Structures
Instructors: Nadeem Kafi, Anum Qureshi, Nida Pervaiz, Basit Jasani	
Student Roll No:	Section No:

Instructions:

- **Attempt SEVEN questions ONLY. Q1 to Q6 are compulsory. Attempt Q7 or Q8, but not both.**
- Read each question completely before answering it. There are **8 questions on 4 pages**.
- You may make assumptions without contradicting any sentence of the respective question.
- **The submitted PDF should list answers according to the sequence given in the question paper.**
- Please follow all the instruction about how to annotate each page of your answer sheet and how to submit your paper online. These have been posted ahead of the exams by (i) your course/lab instructions on the respective Learning Management Systems, (ii) Academic department headed by Mr. Abdul Saeed, and (iii) HOD (CS) office through a general announcement.
- Do not consult any material except this paper in any form using any medium while doing this exam.
- You must take snaps from CamScanner or similar mobile app install on your smart phone. All pages should be snapped in an upright way i.e. portrait mode. There should not be any screen shots in the PDF file which the grader need to rotate to read them properly.
- **You are supposed to attempt this exam alone, in your room located in your parent/guardian's house. No marks will be allocated if this understanding is violated in any way even after submitting the exam and the case will be reported to the department.**
- **Don't share your answers, if your answer is matched to any member of your class, both will get straight F (or U) in the course without asking who shared or who magically copied.**

Time: 3 Hours + 30 minutes for uploading.

Max Points: 100

**Section 1 – Write Code snippets**

Marks: 20 points (20%)

Q1. Write code snippets for the following scenarios.

*Note:*

- No marks will be awarded for text based explanations.*
  - Write useful code only i.e. variables declarations and problem-solving logic. Avoid glue code like #includes, main (), etc. at all costs.*
- Suppose you have received a log file with 200 transactions that was created by an ATM machine. Each transaction consists of six fields: timestamp, card number, amount, 5000 note count, 1000 note count, 500 note count. Store each transaction in an element of a dynamic array. Also, show code snippet that checks that the amount should tally with the note count. [10]
  - Assume that a mobile app stores (i) all incoming 160 characters long SMS messages and (ii) FIVE telephone numbers of family. Suggest a suitable data structure for both data. Now, show code snippet that checks all messages and generate alerts if the message belongs to a family member. [10]

**Section 2 – Short reasoning (No diagram or code here)**

Marks: 20 points (20%)

Q2. Give short and to-the-point (3-4 lines max) reasons for the following. [2x10=20]

- State benefits of chaining over linear probing when implementing hash tables.
- State differences between Kruskal, Prim and Dijkstra algorithms. No write how they function.
- State a scenario where you would avoid the use of linked lists.

- iv. Graph or Tree. Choose and explain which you use to plot a water distribution scheme in cities.
- v. Explain the algorithm you would use to describe the 2<sup>nd</sup> best path in a 5 node graph.
- vi. How sorting can be performed without making comparisons?
- vii. Why would you favor breadth-first search over depth-first search while doing graph traversal?
- viii. State benefits of using a built-in Tree Abstract Data Type (ADT) in your code.
- ix. If the time comes and you have to prefer one over the other, how you justify the time complexity over space complexity and vice versa. Write arguments on both preferences.
- x. Why do you think allocating multiple linked list elements at a time is a good idea? Assume that the linked list is actively used for incoming data.

### Section 3 – Dry run with tracing

**Marks:** 20 points (20%)

Q3. Dry runs the following code snippets. Also trace key variables.

*Note: Marks will be awarded if the grader feels satisfied with your understandings of the code snippets.*

- a) Assume suitable values where initializations are required or if they are provided by the user.

```
void code_snippetONE() {
    char what[80];
    register int p = 0;
    cin.get(what[p]);
    while (what[p] != '\n') cin.get(what[++p]);
    for (p -= 2; p >= 0; cout.put(what[p--]));
}
```

- b) Processing the statement  $s = (w+y)/(t[5]+u)*v$  read from a file; with the algorithm below.

```
delimiterMatching(file)
    read character ch from file;
    while not end of file
        if ch is '(', '[', or '{'
            push(ch);
        else if ch is ')', ']', or '}'
            if ch and popped off delimiter do not match
                failure;
        else if ch is '/'
            read the next character;
            if this character is '*'
                skip all characters until "*" is found and report an error
                if the end of file is reached before "*" is encountered;
            else ch = the character read in;
                continue; // go to the beginning of the loop;
        // else ignore other characters;
        read next character ch from file;
    if stack is empty
        success;
    else failure;
```

#### Section 4 – Scenarios (Do as directed)

Marks: 40 points (40%)

Q4. Imagine A group of friends started a fund raising campaign for needy families. They set stalls at different locations in their city. They also make use of social media platforms to increase the reach. Many people started funding through their program. They have a list of needy families. This list is created with some priorities. Such as, if the family members are greater than 10 in one home, they should be funded earlier and better as compared to the home whose family members are only 3. Many of the needy families register themselves using their stalls. At the end of the day their task is to give priority to the families having more family members.

- Your task is to explain which data structure is suitable for such type of scenario.
- Show step by step, how the following record of families (see Table 1) will get prioritized?

Day1	Fam1	Fam2	Fam3	Fam4	Fam5	Fam6	Fam7	Fam8	Fam9	Fam10
	4	5	2	8	12	3	11	9	10	13

Table 1: Record of families.

Q5. Study the graph (see Figure 1) given below:

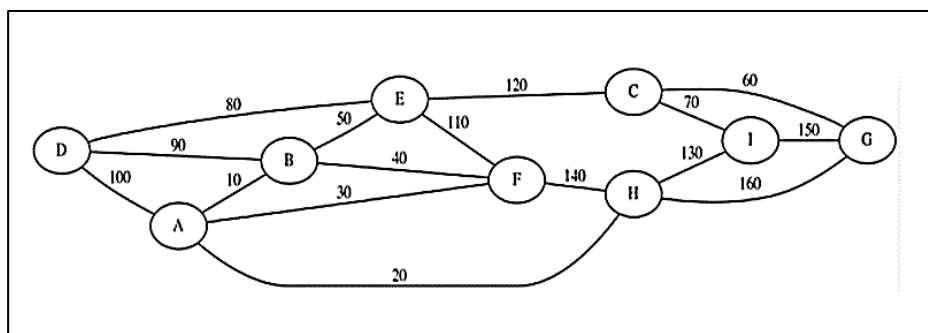


Figure 1: Graph

- In the task **you are required to compute the MST using Prim's algorithm**. Specifically, record the edges of the MST in the request wherein Prim's algorithm adds them to the MST.

A-B

-----

- In the task **you are required to compute the MST using Kruskal's algorithm**. Specifically, record the edges of the MST in the request wherein Kruskal's algorithm adds them to the MST.

A-B

-----

Q6. Consider the hash table with thirteen inserted values in it, as shown in Figure 3.

- Identify the problem associated with the given hash table?
- Provide an example of hash function that could lead to the problem that you identified in part a.
- Provide a better hash function that could resolve the problem.

0	→ 0	→ 30
1		
2	→ 21	→ 31 → 1
3		
4	→ 2	
5		
6	→ 33	→ 63
7		
8		
9		
10	→ 15	
11		
12	→ 26	→ 6
13		
14		
15		
16	→ 38	→ 8
17		
18		
19		

Figure 3: Hash Table

Q7. Implement a city database using a BST to store the database records. Each database record contains the name of the city (a string of arbitrary length) and the coordinates of the city expressed as integer x- and y-coordinates. The BST should be organized by city name. Your database should allow records to be inserted, deleted by name or coordinate, and searched by name or coordinate.

**[Attempt either Question 7 or Question 8, but not both.]**

Q8. Imagine there are lots of people within a room and some of them are having corona. Let's say we have 1000 people overcrowded in this room with having 450 people having corona.

Now our task is to separate corona patients from non-corona patients so that the ones having corona can be shifted to another room and doctors can then deal with these patients accordingly. In order to send these corona patients in some sorted manner by having non-corona patients residing in the same room also having same sorted order. Your task is to explain a step by step procedure that what we should follow in order to achieve the task described above.

Restrictions: We CANNOT send these corona patients to another room one by one, thereby, we need to first sort these corona patients up, then only we can shift all these patients to some other room together.

Following questions should be answered while explaining the procedure. You can have some extra variables associated with each patient.

- a) What sorting criteria do you intend to use and why? This means what value of patient will be taken into consideration for sorting them.
- b) Which sort do you intend to use and why?
- c) How do you intend to separate corona patients from non-corona patients?
- d) What data structure would you prefer to store all corona and non-corona patients separately and why?

----- (X) -----