College of computer
Department of Computer Science

Qassim University جامعة القصيم

وزارة التعليم
كلية الحاسب
قسم علوم الحاسب

**Course: Data Structures / CS214**
**Spring Semester 1445 (2023-24)**

**Mark_____/50**

**Final Exam**
**Duration: 2 hours**
**Thu. 23/05/2023**

| الشعبة: | الرقم الجامعي: | اسم الطالب/ـة: |
|---|---|---|

**For Instructor use only**

| Outcomes | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Course covering Outcomes | √ | √ | | | | |

| Question | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Covered Outcome | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| Allocated mark | 20 | 5 | 4 | 5 | 6 | 5 | 5 |
| Earned mark | | | | | | | |

| Total | | |
|---|---|---|
| 1 | 2 | exam |
| 24 | 26 | 50 |
| | | |

## Question 1: Multiple choice questions [20 marks]

1) What is the time complexity of deleting an element from an array?
   a) O(1) b) O(log n) c) O(n) d) O(n log n)

2) Which of the following represents the in-order traversal of a binary tree?
   a) Left -> Root -> Right b) Left -> Right -> Root
   c) Right -> Left -> Root d) Right -> Root -> Left

3) What is a circular linked list?
   a) A linked list where all nodes are connected in a circular manner.
   b) A linked list where the last node points to the first node.
   c) A linked list where the first node points to the last node.
   d) A linked list where all nodes point to themselves.

4) In a stack, where does a pop operation remove an element from?
   a) At the start of the stack
   b) In the middle of the stack
   c) At the end of the stack
   d) Based on the value of the element

5) In a doubly linked list structure, how many pointers are there in each node?
   a) 1 b) 2 c) 3 d) No pointers

6) What is the advantage of a linked list over an array?
   a) Faster access time.
   b) Less memory usage
   c) Elements can be inserted or removed without reallocation or reorganization of the entire structure.
   d) All of the above

7) Which data structure is used for implementing recursion?
   a) Array b) Linked list c) Queue d) Stack

8) Which of the following is a correct definition of Stack?
a) A data structure that follows the LIFO (Last In First Out) strategy
b) A data structure that follows the FIFO (First In First Out) strategy
c) A data structure that follows both LIFO and FIFO strategies
d) A data structure that doesn't follow LIFO or FIFO.

9) 3) Which of the following is the disadvantage of the array structure?
a) Elements can be accessed sequentially
b) Index of the first element in an array can be negative
c) Stack and Queue data structures can be implemented through an array
d) Wastage of memory if the inserted elements are lesser than the allocated size

10) In an input-restricted dequeue, where does the new element is placed?
a) At the start of the queue
b) At the end of the queue
c) At the start and end of the queue
d) None of the above

11) Any node is the path from the root to the node is called _____.
a) Successor node
b) Ancestor node
c) Internal node
d) None of the above

12) A data structure in which elements can be inserted or deleted at/from both ends.
a) Singe queue
b) Priority queue
c) Dequeue
d) Circular queue

13) Which of the following data structure can provide efficient searching of the elements?
a) Stack
b) Doubly linked list
c) Circular queue
d) Binary search tree

14) The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree?
a) 2 b) 3 c) 4 d) 6

15) What is the time complexity to push an element onto a stack?
a) O(1) b) O(log n) c) O(n) d) O(n log n)

16) _____ of a node is the maximum number of links from root to that node.
a) Height b) Depth c) Degree d) Ancestor

17) Which is the most appropriate data structure for reversing a word?
a) Tree b) Linked list c) Queue d) Stack

18) A binary search tree whose left subtree and right subtree differ in height by at most 1 unit is called _____.
a) AVL tree
b) Complete binary tree
c) Full binary tree
d) Strict binary tree

19) What is the need for a circular queue?
a) Easier computations
b) Effective usage of memory
c) Implement LIFO principle in queues
d) None of the above

20) What is a leaf in a binary search tree?
a) A node with the highest value
b) A node with the lowest value
c) A node with no children
d) None of the above

**Question 2:** State whether the following statements are True (T) or False (F).

**[5 points]**

| | | |
|---|---|---|
| 1. | The size of array has to be pre-decided, while linked lists can change their size any time. | T |
| 2. | If the queue is implemented with a linked list only front pointer will change during an insertion into an non-empty queue | F |
| 3. | The time complexity of searching for an element in a balanced binary search tree is O(n log n). | F |
| 4. | The maximum number of elements a queue can hold depends on the memory available. | T |
| 5. | A strict binary tree is one in which every level, except the last, has the maximum number of nodes, and every node in the last level has only one child, which is its left child. | T |

**Question 3:** Construct a binary search tree of the following elements: **(Q, A, S, I, M, U, N, V, E, R, T, Y)**, and then treaverse all nodes using **Postorder** method.

**[4 points]**

**Question 4:** Implement a function "**push**" using linked list to add a new node to the stack.

**[5 points]**

**Question 5:** Construct an AVL tree when inserting the following values:

20 50 75 98 80 11 13

**[6 points]**

**Question 6:** Write a recursive function in C++ that merges two sorted linked lists into a new sorted linked list and returns the head of new linked list.

**[5 points]**

**Question 7:** Write a function to search for a specific value in the binary search tree and return true if the value is found, false otherwise.

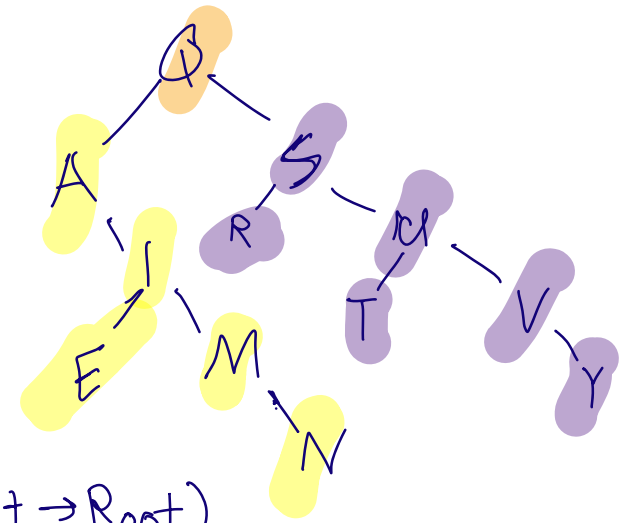
**[5 points]**

*Good Luck!*

Q3

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Q, A, S, I, M, U, N, V, E, R, T, Y




Post (Left → Right → Root)

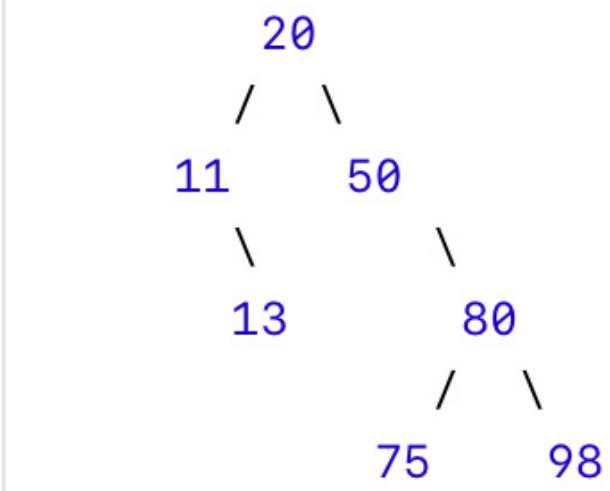
E, N, M, I, A, R, T, Y, V, U, S, Q

Q 4

```
// دالة push لإضافة عنصر جديد
    void push(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = top;
        top = newNode;
        cout << "Pushed: " << value << endl;
    }
```

Q5

نسخ



Q6-7

```
// دمج القائمتين
Node* mergeSortedLists(Node* list1, Node* list2) {
    if (!list1) return list2;
    if (!list2) return list1;

    if (list1->data < list2->data) {
        list1->next = mergeSortedLists(list1->next, list2);
        return list1;
    } else {
        list2->next = mergeSortedLists(list1, list2->next);
        return list2;
    }
}
```

```
// البحث عن قيمة في الشجرة
bool search(Node* root, int value) {
    if (!root) return false;

    if (root->data == value) return true;

    if (value < root->data)
        return search(root->left, value);
    else
        return search(root->right, value);
}
```