Chapter 3 Answers

1. Linked List

2.
   ```
   Step 1:  if (TOS < N-1) Then
   Step 2:      TOS = TOS + 1
   Step 3:      STACK [TOS] = VAL
   Step 4:  Else
   Step 5:      Print "Stack Overflow"
   Step 6:  End of if
   ```

3. Data structure is a particular way of organizing similar or dissimilar logically related data items which can processed as a single unit.

4. FIFO (First In First Out)
5. Stack overflow and underflow:
   - **Overflow**: Occurs when attempting to push an element into a full stack.
   - **Underflow**: Occurs when attempting to pop an element from an empty stack.

6. **Step 1**: Get the address of the first node from Start and store it in Temp.
   Step 2: Using the address in Temp, get the data of the first node and store in Val.
   Step 3: Also get the content of the link part of this node (i.e., the address of the next node) and store it in Temp.
   Step 4: If the content of Temp is not NULL, go to step 2; otherwise stop.

7. Stack Overflow

8. **Step 1**: Check if Stack is full.
   Step 2: If not, Accept the value in variable to insert into stack.
   Step 3: Increment the value of TOS by 1.
   Step 4: Store the value at the TOS position.

9. In linked lists, memory is allocated dynamically at runtime. As long as memory is available in the system, new nodes can be added, so overflow doesn't occur.
10. (a) Stack

11.
    ```
    Step 1:  If (REAR == -1) Then
    Step 2:      FRONT = REAR = 0
    Step 3:      Q[REAR] = VAL
    Step 4:  Else If (REAR < N-1) Then
    ```

```
Step 5:     REAR = REAR + 1
Step 6:     Q[REAR] = VAL
Step 7:  Else
Step 8:     Print "Queue Overflow "
Step 9:  End of If
```

12.

A	B	C
1. stack	ii. Push	d. Removing an item.
2. Queue	i. Front	a. Inserting a new item
3. Array	iv. Subscript	b. Elements are accessed by specifying its position.
4. Linked List	iii. Start	c. Contains the address of the first node.

13. FIFO & LIFO
    - FIFO: First In First Out
    - LIFO: Last In First Out

14. Queue
15. Node
16. Stack
17. A circular queue is a linear data structure in which the last position is connected back to the first to make a circle. It efficiently uses storage and avoids wastage of space in the linear queue during deletions.
18. Operations in data structure:
    - **Traversing**: It is an operation in which each element of data structure is visited once.
    - **Searching**: It is the process of finding the location of a particular element in data structure.
    - **Inserting**: It is the operation in which a new data is added at a particular place in a data structure.
    - **Deleting**: This operation in which a particular element is removed from the data structure.
    - **Sorting**: It is the process of arranging elements in a data structure into ascending or descending order.
    - **Merging**: it is the process of combining elements of two Sorted data structures to form a new one.

19.

```
struct Node {
    string name;
    Node* next;
};
```

20. Queue is a linear data structure which works on the principle of FIFO. Insertion is done at the rear and deletion at the front.

21. Push and Pop operation:
    - **Push**: It is the process of inserting a new data item into stack.
    - **Pop**: It is the process of deleting an element from the top of the stack.

22.

```
Step 1:  If (REAR == -1) Then
Step 2:           FRONT = REAR = 0
Step 3:           Q[REAR] = VAL
Step 4:  Else If (REAR < N-1) Then
Step 5:           REAR = REAR + 1
Step 6:           Q[REAR] = VAL
Step 7:  Else
Step 8:           Print "Queue Overflow "
Step 9:  End of If
```

23. Operations on stack:
    - **Push**: It is the process of inserting a new data item into stack.
    - **Pop**: It is the process of deleting an element from the top of the stack.