DSA Lab 1

Q1. Write a program to accomplish each of the following with respect to the given structure:

```
typedef struct student{
        char name[25];
        int age;
        float marks[5];
        float avgMarks;
}Student;
```

1. Use Student to declare variable **s1** to be of type struct student, array **students[10]** of type struct student and variable **sptr** to be of type pointer to struct student.
2. Read **name, age** and **marks** from keyboard and calculate **avgMarks** (of struct student) based on the input **marks** for the following
    a. **s1**
    b. **sptr**
    c. 10 elements of array **students**
3. Calculate average of **avgMarks** of all ten elements of array **students** and print **name** of those elements who have **avgMarks** more than the calculated average of all ten elements.

Q2. Given two sorted arrays arr1 and arr2 of sizes n1 and n2 respectively, merge them to get the final sorted array arr3. Don't use any sorting algorithm for this problem. What will be the time complexity in terms of Big O.

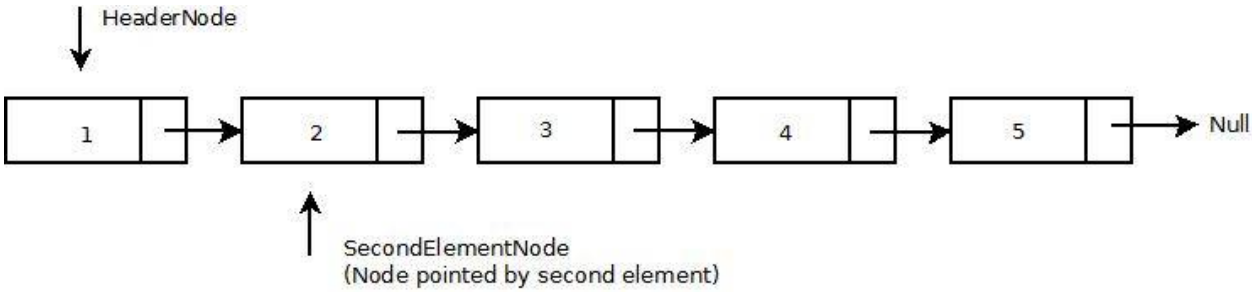
Q3. Reverse the character string using stack. Implement stack in form of a structure.

Q4. Given a sorted array arr1, find a query element using binary search.

Data Structure: Singly Linked list

Singly linked list is a basic linked list type. Singly linked list is a collection of nodes linked together in a sequential way where each node of singly linked list contains a data field and an address field which contains the reference of the next node.

To perform any operation on a linked list we must keep track/reference of the first node which may be referred by **head pointer variable**. In singly linked list address field of last node must contain a NULL value specifying end of the list.



Basic structure of a singly linked list

Each node of a singly linked list follows a common basic structure. In a node we can store more than one data fields but we need at least single address field to store the address of next connected node.

```
struct node {
    int data;           // Data
    struct node * next; // Address
};
```

Advantages of Singly linked list

There are several points about singly linked list that makes it an important data structure.

- Insertion and deletion of elements doesn't require movement of all elements when compared to an array.
- Can allocate or deallocate memory easily when required during its execution. (Dynamic Memory allocation)

Disadvantages of Singly linked list

- Linked Lists are sequential access while Arrays are random access.
- Extra memory space is required in Linked list for pointers.

DSA Lab Assignment#2 (Batch:Monday 10.00-12.00)

Write C Code for the following:

Q1. To create a Singly Linked List to store positive integers in the order they are inputted.

Q2. To count the number of nodes in a Singly Linked List.

Q3. To split the Singly Linked List (Created in Q1) in two linked list (List-1 should contain only even values while list-2 contains only odd values).

Q4. To display all nodes of a given Singly Linked List. (Traversal of Singly Linked List)

Lab Assignment#3

Objective: The objective of this lab is to move ahead in performing various operations on Linked Lists.

Write the programs using C for the following:

1. To find the middle of a given Linked List.

   a. Delete this identified middle element.

2. To create a sorted linked list.

3. To find nth node from the end of the linked list.

Optional question

4. Remove the duplicate elements from a sorted linked list.

Lab Assignment#4

Objective: The objective of this lab is to cover the implementation and application of Stacks.

Write C code for the following:

1. Implement the stack operations (push (), pop (), top (), is empty () etc) using linked list.
2. Use the Stack to convert any infix expression into postfix.
3. Use the stack to evaluate postfix expression.

Optional question

1. Use stack to check that every opening bracket has its respective closing bracket.

Set-1

Q1. Write C code to implement stack using arrays.

Q2. A stack contains some integer numbers in unsorted order. Sort these elements in ascending order using other stacks.

Set2.

Q1. Write C code to implement queue using Linked list.

Q2. Given a number n, write a function that generates and prints all binary numbers from 1 to n (Use Queue).

Input: n = 5
Output: 1, 10, 11, 100, 101

Assignment#5 Tuesday (Morning 10.00 – 12.00 Noon)

Write C code for the following:

Q1. To implement two stacks using a Single Array.

Q2. Vikas inserted some number blocks in a bag(as Stack). Vikas has to organize the blocks again in the same order as he has inserted it into the bag, **i.e. the first number inserted into the bag by Vikas should be picked up first followed by other numbers in series.** Help Vikas to complete this work in O(n) time complexity with the condition to use one extra space O(1).

Lab Assignment#5 Tuesday Lab(Afternoon, 4.00-6.00)

Q1. You are given a stack of integers of size **N**. pop an element from the stack or push any popped element into the stack is counted as one operation. Find the maximum of the stack after performing exactly **K** operations. If the stack becomes empty after performing K operations and there is no other way for the stack to be non-empty, print **-1**.

Q2. Traffic lights have **bulbs** on the traffic board and only when one of them is green(**G**) the cars can pass. there are **2** other states also which the bulb can show; i.e. Red(**R) &** Yellow(**Y**). Note that the lights are designed such that they follow a state change cyclic pattern as follows:

R------>Y------>G------->R

Initialize all the bulbs light to any state and demonstrate the behavior of traffic light after every second. (Hint use Queue).

Q1. Given a stack, the task is to sort it such that the top of the stack has the greatest element.

Q2. Given an integer N, remove consecutive repeated digits from it. For example, if the input is 1222445, the output should be 1245

Q1. Given a sequence of n strings, the task is to check if any two similar words come together then they destroy each other then print the number of words left in the sequence after this pairwise destruction.

Input: ab aa aa bcd ab
Output: 3
As aa, aa destroys each other so, ab bcd ab is the new sequence.

Input: tom jerry jerry tom
Output: 0
As first both jerry will destroy each other. Then sequence will be tom, tom they will also destroy each other. So, the final sequence doesn't contain any word.

Q2. Given an integer N, remove consecutive repeated digits from it. For example, if the input is 1222445, the output should be 1245

Q1. WAP to implement two Queue using single Array.

Q.2 Vikas inserted some number blocks in a bag(as Stack). Vikas has to organize the blocks again in the same order as he has inserted it into the bag, **i.e. the first number inserted into the bag by Vikas should be picked up first followed by other numbers in series.** Help Vikas to complete this work in O(n) time complexity with the condition to use one extra space O(1).

Assignment#5 DSA Lab Thursday 14_02_2019 (4.00-600 PM)

Q1. Implement Deque using circular array

Deque or Double Ended Queue is a generalized version of Queue data structure that allows insert and delete at both ends.
Operations on Deque:

1. insertFront(): Adds an item at the front of Deque.
2. insertRear(): Adds an item at the rear of Deque.
3. deleteFront(): Deletes an item from front of Deque.
4. deleteRear(): Deletes an item from rear of Deque.

In addition to above operations:

1. getFront(): Gets the front item from queue.
2. getRear(): Gets the last item from queue.
3. isEmpty(): Checks whether Deque is empty or not.
4. isFull(): Checks whether Deque is full or not.

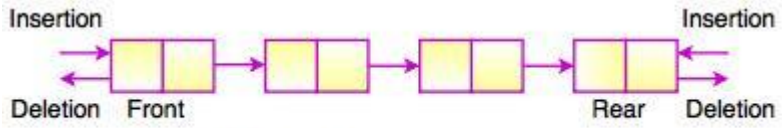


Fig. Double Ended Queue (Dequeue)

Q2. Implement a stack which will support three additional operations in addition to push( ) and pop( ):

1. peekLowestElement()

   //return the lowest element in the stack without removing it from the stack

2. peekHighestElement()

   //return the highset element in the stack without removing it from the stack

3. peekMiddleElement()

   //returns the middle element in the stack without removing it from the stack.

Time complexity of each operation should be O(1).