

DATA STRUCTURES QUESTIONS (LAB)

1. Students of a Programming class arrive to submit assignments. Their register numbers are stored in a LIFO list in the order in which the assignments are submitted. Write a program using array to display the register number of the ten students who submitted first.

Register number of the ten students who submitted first will be at the bottom of the LIFO list. Hence pop out the required number of elements from the top so as to retrieve and display the first 10 students.

Low Level: Display the register number of the last submitted record

Middle Level: Display the register number of the ten students who submitted first

High Level: Should implement low level and middle level modules + any query posed by faculty such as checking if a particular student has submitted the assignment or not

2. To facilitate a thorough net surfing, any web browser has back and forward buttons that allow the user to move backward and forward through a series of web pages. To allow the user to move both forward and backward two stacks are employed. When the user presses the back button, the link to the current web page is stored on a separate stack for the forward button. As the user moves backward through a series of previous pages, the link to each page is moved in turn from the back to the forward stack.

When the user presses the forward button, the action is the reverse of the back button. Now the item from the forward stack is popped, and becomes the current web page. The previous web page is pushed on the back stack. Simulate the functioning of these buttons using array implementation of

Stack. Also provide options for displaying the contents of both the stacks whenever required.

Low Level: Implement either forward stack or backward stack

Middle Level: Implement web browser navigation using both the stacks

High Level: Use a single array to implement both the stacks

3. Design a program to employ a stack for balancing symbols such as parentheses, flower braces and square brackets, in the code snippet given below.

```
for(i=0;i<n;i++)
```

```
{
```

```

if(i<5)
{
    z[i]=x[i]+y[i];
    p=(((a+b)*c)+(d/(e+f)*g);
}

```

Ensure that your program works for any arbitrary expression.

Low Level: Implement balancing of parenthesis only

Middle Level: Implement balancing of all symbols only for some specified expressions only
c

high Level: Implement for any arbitrary expression given by faculty

4. Most of the bugs in scientific and engineering applications are due to improper usage of precedence order in arithmetic expressions. Thus it is necessary to use an appropriate notation that would evaluate the expression without taking into account the precedence order and parenthesis.

a) Write a program to convert the given arithmetic expression into

i) Reverse Polish notation

ii) Polish notation

b) Evaluate the above notations with necessary input.

Low Level: Implement one conversion and its evaluation

Middle Level: Implement both polish and reverse polish conversions

High Level: Implement both a and b

5. Some priests are given three poles and a stack of 4 gold disks, each disk a little smaller than the one beneath it. Their assignment is to transfer all 4 disks from one of the 3 pole to another with 2 important constraints. They can move only one disk at a time, and they can never place a larger disk on top of a smaller one. Design a recursive program for the above Towers of Hanoi puzzle using stack.

Low Level: Implement the problem using recursion

Middle Level: Implement the problem using recursion and also trace the flow of execution

High Level: Implement without recursion at least for few disks.

QUEUE ADT:

6. In a theme park, the Roller-Coaster ride is started only when a good number of riders line up in the counter (say 20 members). When the ride proceeds with these 20 members, a new set of riders will line up in the counter. This keeps continuing. Implement the above scenario of lining up and processing using arrays with Queue ADT.

Low Level: Implement the aforementioned problem [6 Marks]

Middle Level: Also count the number of adults and children [2 Marks]

High Level: If a VIP family arrives, process them before processing others who are already waiting in the queue [2 marks]

7. When burning a DVD it is essential that the laser beam burning pits onto the surface is constantly fed with data, otherwise the DVD fails. Most leading DVD burn applications make use of a circular buffer to stream data from the hard disk onto the DVD. The first part, the 'writing process' fills up a circular buffer with data, then the 'burning process' begins to read from the buffer as the laser beam burns pits onto the surface of the DVD. If the buffer starts to become empty, the application should continue filling up the emptied space in the buffer with new data from the disk. Implement this scenario using Circular Queue.

Low Level: Implement the above problem [6 marks]

Middle Level: Also print the front and rear indices at any instant [2 marks]

High Level: At any instant, individually display the data which has been inserted newly and the old data that has not yet been burnt [2 marks]

8. a) There is a garage where the access road can accommodate any number of trucks at one time. The garage is built in such a way that only the last truck entered can be moved out. Each of the trucks is identified by a positive integer (a truck_id). Implement dynamically to handle truck moves, allowing for the following commands:

i) On_road (truck_id); ii) Enter_garage (truck_id);

iii) Exit_garage (truck_id); iv) Show_trucks (garage or road);

If an attempt is made to get a truck out which is not the closest to the garage entry, the error message “Truck x cannot be moved” should be displayed.

b) For the aforementioned scenario, assume now a circular road and two entries: one for entry, another for exit. Trucks can get out only in the order they got in. Write a program dynamically to handle truck moves allowing for the following commands

i) Enter garage (truck name)

ii) Exit garage (truck name)

iii) Show trucks

Low Level: Implement the above problem [6 marks]

Middle Level: Also check the number of services done so far to decide if service charge is required or not [2 marks]

High Level: Assuming any n^{th} truck will need a longer period of time for service, that particular truck should be retained in the garage and the links in the stack should be modified accordingly [2 marks]

LIST ADT

9. Details such as Register number, name, quiz-3 marks of those students who wrote for SET-question paper and of those who wrote for SET-B question paper are maintained in sorted order in two separate lists L1 and L2. Write a program to merge the two lists in sorted order so as to facilitate mark entry in faculty login.

Low Level: Merging list 2 at the end of list 1 [6 marks]

Middle Level: Implement the above problem [2 marks]

High Level: Enhance the above problem for unsorted lists [2 marks]

10. Write a program to maintain the records of students in an effective dynamic structure. Search a particular record based on the roll number and display the previous and next values of that node with time complexity of $O(1)$.

Low Level: Implement the above problem [6 marks]

Middle Level: Delete user specified record and then display the current predecessor and successor [2 marks]

High Level: Also for the first node make the last node as the predecessor and for the last node make the first node as the successor and thus make it a circular doubly linked list.

[2 marks]

11. Assume FLAMES game that tests for relationship has to be implemented using a dynamic structure. The letters in the FLAMES stand for Friends, Love, Affection, Marriage, Enmity and Sister. Initially store the individual letters of the word 'flames' in the nodes of the dynamic structure. Given the count of the number of uncommon letters in the two names 'n', write a program to delete every nth node in it, till it is left with a single node. If the end of the dynamic structure is reached while counting, resume the counting from the beginning. Display the letter that still remains and the corresponding relationship

Eg., If Ajay and Jack are the two names, there are 4 uncommon letters in these. So delete 4th node in the first iteration and for the next iteration start counting from the node following the deleted node.

Low Level:Delete only the first nth letter only [6 marks]

Middle Level:Implement the above problem [2 marks]

High Level: For the same problem instead of deleting the nth letter, make the nth letter as the last node. Hence at last the first node gives the relationship [2 marks]

SORTING AND SEARCHING

12. Assume in the Regional Passport Office, a multitude of applicants arrive each day for passport renewal. A list is maintained in the database to store the renewed passports arranged in the increased order of passport ID. The list already would contain there cords renewed till the previous day. Apply Insertion sort technique to place the current day's records in the list.

Later the office personnel wish to sort the records based on the date of renewal so as to know the count of renewals done each day. Taking into consideration the fact that each record has several fields (around 25 fields), follow Selection sort logic to implement the same.

Low Level: Implement one of the above scenarios [6 marks]

Middle Level: Implement the problem for both the scenarios [2 marks]

High Level: Also your program should check the stability of both the algorithms [2 marks]