

Lab Manual

Course No.: 1202

Title: Data Structure Lab

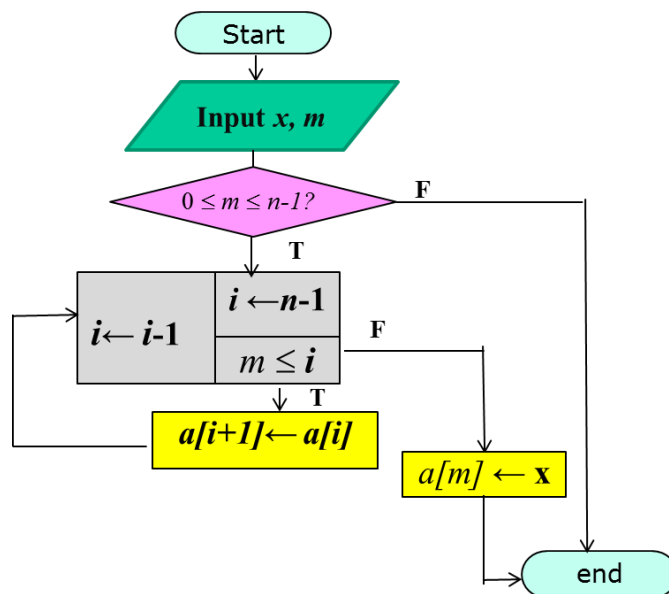
Instructure: Md. Shahid Uz Zaman

Report: Submit report for each lab including flowcharts, programs, outputs, explanation and comments

Lab 1(Week 1, 2A/2B/2C/2D): Based on Arrays

Prob1: Write a C program to test the following algorithm regarding inserting new elements in an array.

Hints: First create an array with several inputs then try to insert a new element



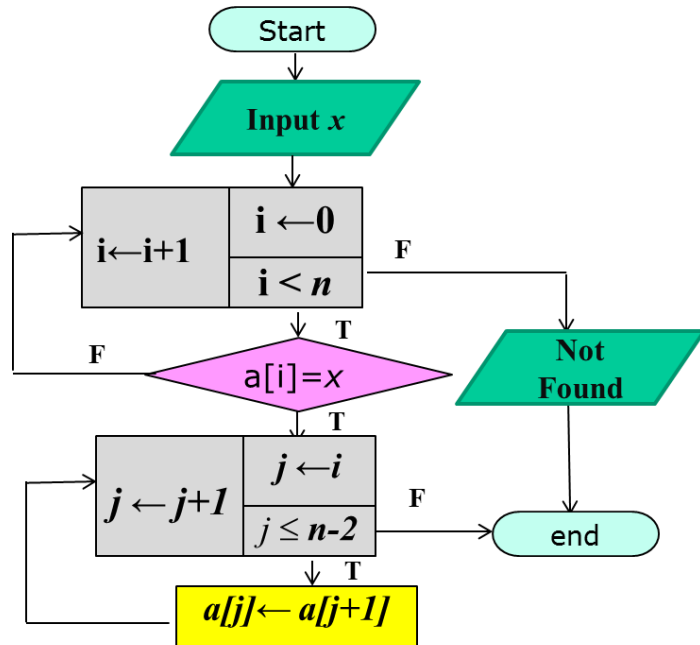
n : total elements
 m : index $0 \leq m \leq n-1$
 x : input variable for new data
Array Elements: $a[0] \dots a[n-1]$

Output: print the array after each new insertion. Also calculates and prints the no. of shifting required for each insertion.

Update: Modify the algorithm so that it can display the appropriate messages like “array is full” when there will be no space in the array. Also modify it for inserting several elements in a single run.

Prob2: Write a C program to test the following algorithm regarding deleting an existing array element.

Hints: First create an array with several inputs then try to delete a new element.



n: total elements
m: index $0 \leq m \leq n$
x: input variable to be deleted
 Array Elements: $a[0] \dots a[n-1]$

Output: print the array after each deletion. Also calculates and prints the no. of shifting required for each deletion.

Update: Modify the algorithm so that it can display the appropriate messages like “**array is empty**” when there will be no element left to be deleted in the array. Also modify it for deleting several elements in a single run.

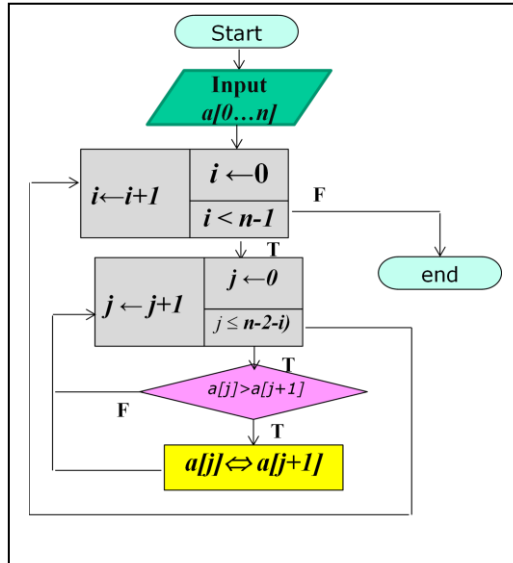
Prob3: Write an algorithm and its corresponding C program to combine the above two algorithms for insertion and deletion in the same time using the following menu.

Hints: Start the algorithm with empty array then try to insert and delete using following menu. The program will ends with user selects option 4.

1. Insert
 2. Delete
 3. Display
 4. Exit
- Enter your option(1-4):

Lab 2 (Week 2, 3A/3B/3C/3D): Based on Arrays

Prob1: Write a C program to sort n numbers using Bubble Sort Method.



Input: 10, 30, 70, 16, 95, 22

Remarks: Write the content of the array after each iteration.

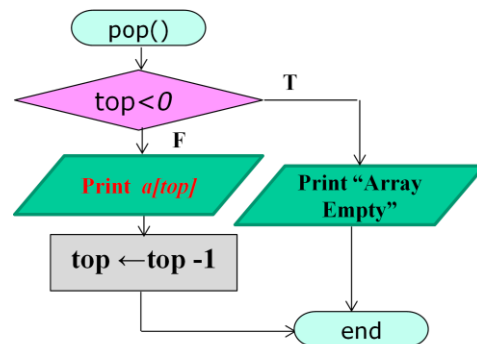
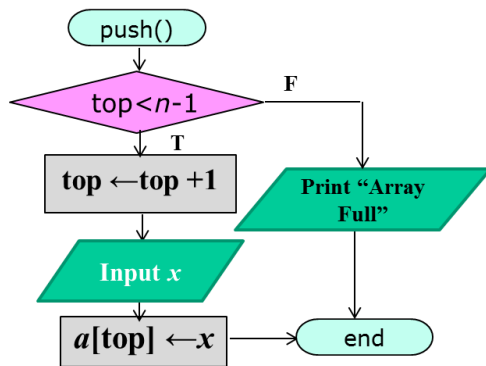
Prob 2: Write an algorithm to merge two sorted array into one sorted array.

Hints: First input elements to two arrays and sort them. And then merge them to get one sorted array.

Lab 3 (Week 3, 4A/4B/4C/4D): Based on Stacks

Prob1: Write an algorithm to implement a stack using PUSH and POP methods. Also write a C program to test the algorithm written using the following menu.

1. Push
 2. Pop
 3. Display
 4. Exit
- Enter your option(1-4):

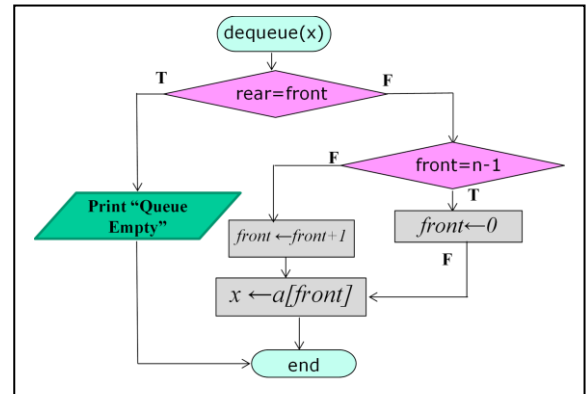
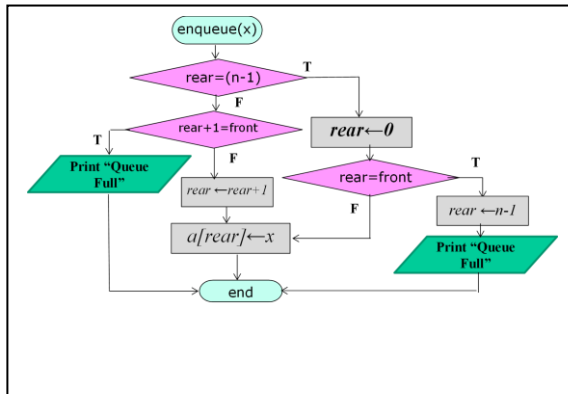


Remarks: Test the program using all the boundary values

Lab 4 (Week 4, 5A/5B/5C/5D): Based on Queues

Prob1: Write an algorithm to implement a Queue using Enqueue and Dequeue methods. Also write a C program to test the algorithm written using the following menu.

1. Enqueue
 2. Dequeue
 3. Display
 4. Exit
- Enter your option(1-4):

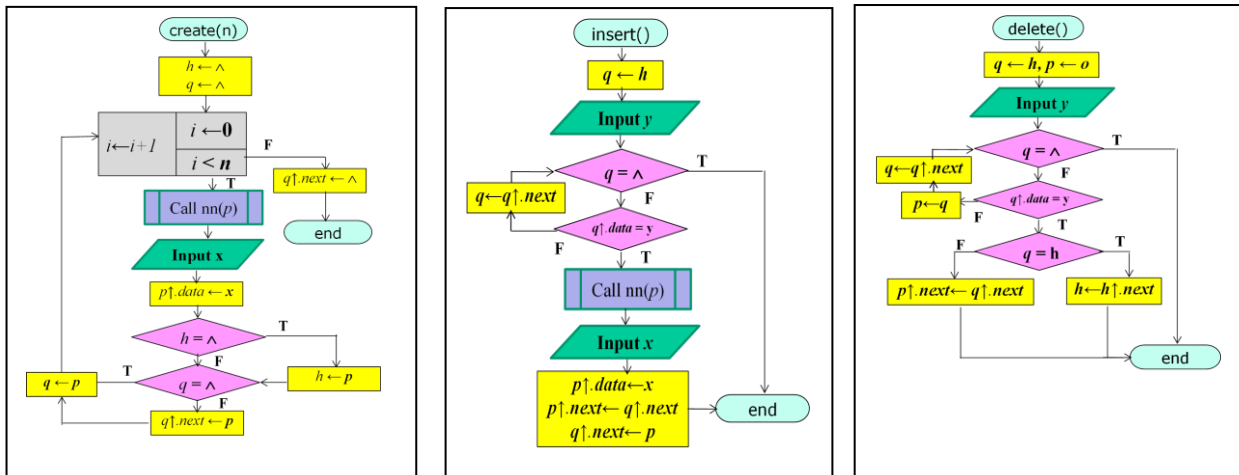


Lab 5 (Week 5, 6A/6B/6C/6D): Based on Single Linked List

Prob1: Write an algorithm to **CREATE** a single linked list with n nodes. Also write a C program to test the algorithm written using the following menu.

Prob2: Write an algorithm to **INSERT** a new node after a existing node in a single linked list. Also write a C program to test the algorithm written using the following menu. Hint. Use the created Linked List which is created in Prob 1:

Prob3: Write an algorithm to **DELETE** a particular node of a single linked list. Also write a C program to test the algorithm written using the following menu. Hint. Use the created Linked List which is created in Prob 1:



Prob1: Combine the previous three algorithms (1-3) to create a menu program to automate the manipulation of single linked list. Also write a C program to test the algorithm written using the following menu.

1. Create
 2. Insert
 3. Delete
 4. Display
 5. Exit
- Enter your option(1-5):

Remarks: Test the program using all the boundary values

Lab 6 (Week 6, 7A/7B/7C/7D): Based on Circular Linked List

Prob1: Use **CREATE**, **INSERT** and **DELETE** algorithms to create a menu program to automate the node manipulation of single circular linked list. Also write a C program to test the algorithm written using the following menu.

1. Create
 2. Insert
 3. Delete
 4. Display
 5. Exit
- Enter your option(1-5):

Remarks: Test the program using all the boundary values