

## WEEK-5 (17 Feb - 22 Feb 2020)

### CO 2

## Lab Test- I (24 Feb to 29 Feb 2020)

### (Stack based solutions for data storage, retrieval, searching, and sorting problems)

- Q 1. How can you implement two stacks in a single array, where no stack overflows until any space left in the entire array space?
- Q 2. Write a program in C to implement priority Queue using arrays. A typical priority queue supports following operations.  
**insert(item, priority):** Inserts an item with given priority.  
**getHighestPriority():** Returns the highest priority item.  
**deleteHighestPriority():** Removes the highest priority item.
- Q 3. Write a program in C to implement D queue using arrays. **Operations on Deque:**  
Mainly the following four basic operations are performed on queue:  
**insetFront():** Adds an item at the front of Deque.  
**insertLast():** Adds an item at the rear of Deque.  
**deleteFront():** Deletes an item from front of Deque.  
**deleteLast():** Deletes an item from rear of Deque.  
  
In addition to above operations, following operations are also supported  
**getFront():** Gets the front item from queue.  
**getRear():** Gets the last item from queue.  
**isEmpty():** Checks whether Deque is empty or not.  
**isFull():** Checks whether Deque is full or not
- Q 4. Write a function called copy\_stack that copies the contents of one stack into another. The function must have two arguments of type stack, one for the source stack and one for the destination.
- Q 5. Write a program in c to convert infix notation to postfix notation.
- Q 6. Write a program to check on validity in an expression by matching opening and closing parenthesis.
- Q 7. Write a program to reverse stack using recursion.
- Q 8. Write a recursive program to insert an element in the sorted array using stack.
- Q 9. a) Write a program that read numbers from a file and pushes them into stack until we read a negative no. At this time, we stop reading and pop five items from the stack and print them. If there are fewer than five items in the stack, print the error message. After printing data

resume reading data and placing them in the stack. When the end of file is detected, print a message and item remaining in the stack.

b) Write a program that prints the five numbers preceding a negative no as described. If there are lower than five numbers, print the numbers that exists.

**Q 10.** Find the output of the following questions.

<pre>#include&lt;stdio.h&gt; void fun(int x) {     if(x &gt; 0)     {         fun(--x);         printf("%d\t", x);         fun(--x);     } } int main() {     int a = 4;     fun(a);     return 0; }</pre>	<pre>int fun1(int x, int y) {     if(x == 0)         return y;     else         return fun1(x - 1, x + y); } int main() {     int arr[] = {12, 10, 30, 50, 100};     printf(" %d ", fun(arr, 5));     getchar();     return 0; }</pre>
--	--

Recursive function to insert an item x in sorted way

```
void sortedInsert(struct stack **s, int x)
{
    // Base case: Either stack is empty or newly inserted
    // item is greater than top (more than all existing)
    if (isEmpty(*s) || x > top(*s))
    {
        push(s, x);
        return;
    }

    // If top is greater, remove the top item and recur
    int temp = pop(s);
    sortedInsert(s, x);

    // Put back the top item removed earlier
    push(s, temp);
}

// Function to sort stack
void sortStack(struct stack **s)
{
    // If stack is not empty
    if (!isEmpty(*s))
    {
        // Remove the top item
        int x = pop(s);
```

```
// Sort remaining stack
sortStack(s);

// Push the top item back in sorted stack
sortedInsert(s, x);
}
```