National University of Computer and Emerging Sciences

Laboratory Manual 07

for

Data Structures Lab

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

- Recursion
- Queue

Q1. Given an encoded string, return its decoded string.
The encoding rule is: k[encoded_string], where the encoded_string inside the square brackets is being repeated exactly k times. Note that k is guaranteed to be a positive integer.
You may assume that the input string is always valid; there are no extra white spaces, square brackets are well-formed, etc. Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, k. For example, there will not be input like 3a or 2[4].
The test cases are generated so that the length of the output will never exceed 105.

Example 1:
Input: s = "3[a]2[bc]"
Output: "aaabcbc"
Example 2:
Input: s = "3[a2[c]]"
Output: "accaccacc"
Example 3:
Input: s = "2[abc]3[cd]ef"
Output: "abcabccdcdcdef"

Q2: Find all the r Combinations of an array of size n using recursion.

Input: arr=[1,2,3,4], r=2

Output: 1 2
1 3
1 4
2 3
2 4
3 4

Input: arr=[1,2,3,4], r=3

Output: 1 2 3
1 2 4

1 3 4
2 3 4

QUEUE

Question 3

Implement a template-based queue using a fixed-sized array. The required member methods are:

int size() : returns the count of total element stored in the queue.
bool isEmpty(): returns true if the queue is empty else false.
bool front(T&): returns, but does not delete, the front element fromthe queueviatheparameter passed by reference. It returns false if there is no element in the queue, elseitreturns true and assigns the front element of the queue to the parameter passedbyreference.
void dequeue(): deletes the front element from the queue. If there is no element, returnsome error.
void enqueue(T const& e): inserts the element “e” at the back of the queue if thereissomespace available. Otherwise it returns some error.
Int secondhighest() find the second highest element of a queue.
Void intersection()find the intersection of two queues.