

Department of Computer Science

CS 201 – Data Structures

Mid Term I (Fall 2013)

Instructors: Adnan Waheed & Rabia Maqsood

September 23, 2013

Total Marks: 40	Time Allowed: 60 minutes
------------------------	---------------------------------

Instructions:

- (1) Understanding the question is part of exam. NO QUERIES WILL BE ENTERTAINED.
- (2) Provide answers for Q#1 in the space given and all the rest on answer sheet.
- (3) Write neat & clean.
- (4) Use permanent ink pens only.
- (5) Poor programming approaches will decrease your marks.
- (6) Think about the boundary conditions
- (7) Comment your code, your quiz won't be evaluated otherwise
- (8) Poor programming approaches will decrease your marks
- (9) There should be no memory leaks nor dangling pointers

Roll No. _____ Name: _____ Section: _____

Question No.	1	2	3	Total
Marks	20	10	10	35

GOOD LUCK 😊



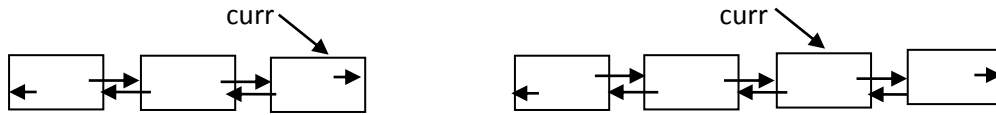
Question 1:

Marks 20 (5 each)

1. Write a procedure *stackCapacity* to obtain the capacity of a linked stack represented by its top pointer TOP. The capacity of a linked stack is the number of elements in the list forming the stack.

2. Write a function *isLast* which returns true if the given node is the end node in a linked list and false otherwise.

3. Write a single code that deletes the node pointed by curr in the given 2 cases of a doubly linked list.



4. Part – a is for sections C & D however sections A & B should solve Part – b for this question.
- a. Convert the given postfix expression into equivalent infix expression. Show all steps, no marks will be given for direct answer.

$a\ b\ +\ c\ -\ d\ e\ *\ /$



- b. Write a function *copyStack* which transfer elements from stack S1 to stack S2 so that elements of stack S2 are in the same order as on S1 using one additional stack.

Question 2:**Marks 10**

Consider two sorted linked lists L1 and L2, both containing integers in increasing order. Merge these two lists into one list L3 such that L3 should also be sorted but in decreasing order. For example L1 (1, 3, 5, 6) and L2 (2, 4, 8, 9) are merged into a list L3 (9, 8, 6, 5, 4, 3, 2, 1). We do not know the size of lists L1 and L2. Write a generic piece of code to handle all cases.

Question 3:**Marks 10**

A Queue ADT is implemented using an array $a[\text{MaxSize}]$ and two indices, *Front* and *Rear*. *Front*, which holds the index of the front element of the queue, is free to move. *Rear*, which holds the index of the rear element of the queue, is fixed at $\text{MaxSize}-1$. Given these constraints, implement the function `Enqueue (int val)` for this ADT. If you use other functions, please also specify them completely.