



Çankaya University

Department of Computer Engineering

CENG 218 Data Structures

Instructor: Abdulkadir GORUR

Time Allowed: **120** minutes

Date: 14 / 04 / 2008

Name and Surname :

Student Number :

Q1. Evaluate the following postfix expression, showing the values in the stack at each indicated point in the Postfix string (points A, B, and C). (2p for each point and 2p for final value)

8	5	*	4	6	+	-	2	8	*	-	4	5	+	*
					A				B				C	
A					B				C					

The Final value of the expression is _____

Q1. Write a function element with parameter k , which returns the value of the k th element on the stack of integers. For instance, if k is 1, it returns the value of the current first element. It does not disturb the stack at all. Assume you do not have access to the stack directly, but only through the push, pop, isEmpty and isstackFull functions **(10p)**

Q2. Write a C program to read a group of input lines, each containing one word. Print each word that appears in the input and the number of times that it appears. (word "END" terminates the input process) (25p)

Q3. Consider a linked list whose members are integer numbers only. (20 P)
Write a function ***removeitems(NODE *head)*** function that removes the **single digit** elements from the linked list. Consider following example:

Input LList:Listhead-> 3-> 41-> 3-> 41-> 2-> 7-> 7-> 1-> 9->12-> NULL

Node * removeItems(NODE *head) {

Q3. write a function which would delete the alternate nodes starting with the second node, and would return the remaining list. It is specified that the given list, pList contains at least two nodes. However, total number of nodes could be even or odd. Your code should be able to handle both the cases. (20p)

Q4. Write a function called **compressString(s1,s2)** that takes two strings as its arguments. The String s1 contains null terminated characters. s1 may contain repeating alpha numeric characters. The number of repetition may be between 1 and 9. Write a function that compresses s1 in as the number of repetition and the repeating character.

Following illustrates how **compressString** function generates s2 based on s1.(assume s2) (25p)

Ex1: s1="aaaaaezzzzbbb"
 S2="4a8e3z4b"

Ex2: s1="aallliiii"
 s2="2a3l5i"

Ex3: s2="t66kkkqqqq"
 s1="1t263k5q"

Q4. **(15p)** write a function **replaceStr(char *str,char *olds,char *news)** that replaces every occurrence of old string in str with news. (assume olds and news are same in length)

Q5. Write the function *Replicate* whose header is given below. *Replicate* adds new nodes to *list* so that nodes are replicated the number of times specified by parameter *count* (25p).

For example, if linked *list* is represented by
 "ayse", "data", "xray"

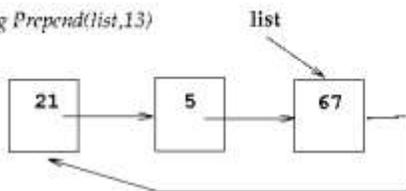
Then the call *Replicate*(*list*,3) should change *list* as shown below.

" ayse ", " ayse ", " ayse ", " data ", " data ", " data ", "xray", "xray", "xray"

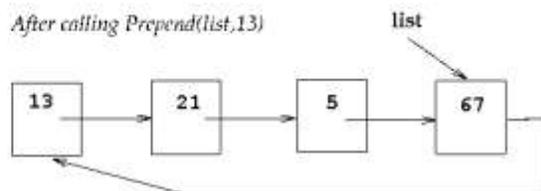
void Replicate(Node * list,int count)

Q5. A circularly-linked list is maintained by keeping a pointer to the last node: the first node is then the node "after" the last node (since the list is circular). Write two functions: *Prepend* that adds a new node to the front of a circularly-linked list and *RemoveLast* that removes the last node from a circularly-linked list. (4p for *Prepend* 6p for *RemoveLast*)

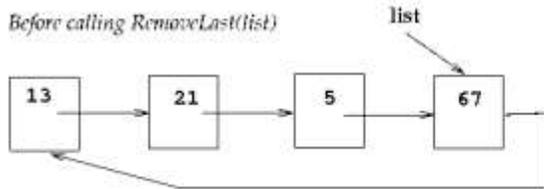
Before calling *Prepend*(*list*,13)



After calling *Prepend*(*list*,13)



Before calling *RemoveLast*(*list*)



After calling *RemoveLast*(*list*)

