



Çankaya University

Department of Computer Engineering

CENG 218 Data Structures Final

Instructor: Abdulkadir GORUR

Time Allowed: **120** minutes

Date: 23 / 05 / 2008

Name and Surname :

Student Number :

Q1. (15p) Given binary tree struct definition:

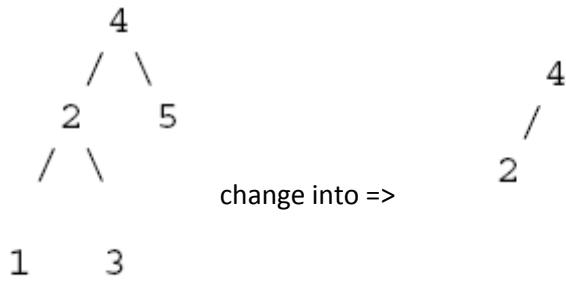
```
typedef struct tnode{
    int val;
    struct tnode *lchild;
    struct tnode *rchild;
} TNODE ;
```

Write the recursive function **lookup** which returns 1 if "value" is contained in right sub tree of root, 2 if value is contained in the left sub tree of the root. 0 if value is contained in root. -1 otherwise.

- Q2. (10p) Write the implementations of the **PUSH** and **POP** functions, where the stack is implemented as a **singly linked list**. (assume create node function and isEmpty functions are available).
- Q2. (15p) Write the implementations of the **PUSH** and **POP** functions, where the stack is implemented as a **doubly linked list**. (assume create node function and isEmpty functions are available).
- Q2. (20p) Write the implementations of the **enqueue** and **dequeue** functions, where the queue is implemented as a **singly linked list**. Use the following node and stack class declarations. (assume create node function and isEmpty functions are available).

Q3. (10p) Suppose that you have a stack *ADT*. Write a function(*delAll(STACK *s,int item)*) that deletes every occurrence of a specified item from *S*, leaving the order of the remaining elements unchanged.

Q4. (15p) Write a function that will take the root of a binary tree and will remove all leaf nodes of tree.



Q4. (15p) Write a function that returns parent node of a given node **node** in binary tree. The function prototype is as follows (NULL pointer will be returned either if node is not in binary tree or node is root)
TNODE *findParent(TNODE *root, TNODE *node)

Q5. (10P) Write a function **NODE *moveToLast(int key, ElementType key)** that searches list L for a node that matches a key value. If key exists in L search causes the node to be moved to the tail of the list. An unsuccessful search leaves the list unchanged. For example, if L={1,2,3,4}, then moveToLast(3,L) produces L={1,2,4,3}.

```
typedef struct node{  
    ElementType x;  
    struct node *next;  
} NODE;
```

Q6. (20 P) Consider a linked list whose members are integer numbers only.

Write a function ***removeItems(NODE *head)*** function that removes the **odd** valued elements. Consider following example:

Input LList:Listhead->5->3-> 4-> 3-> 4-> 2-> 7-> 7-> 1-> 9->12-> NULL

Output LList:Listhead-> 4 -> 4 -> 2 -> 12 -> NULL

Node * removeItems(NODE *head){

Q7. (20p) Recall that the ADT list operations are;

- void createList(LIST *)
- int isEmpty(LIST *)
- int size(LIST *)
- void add(LIST *,int item, int index)//inserts item at specified position (first position is 1)
- void remove(LIST*,int index)//removes item from specified position
- void removeAll(LIST*)
- int indexOf(LIST *,int item)//returns the index of item
- int itemAt(LIST *,int index)//returns the item in position specified by index

Implementation of LIST ADT operations and details are hidden. Only the methods listed above are available for use.

Assume list contains some integer numbers and this list may contain repeated consecutive integer numbers.

Write a function **removeRepeated(LIST *)** to remove those repeated consecutive integer numbers from the list by leaving only one copy of them.

Consider following example:

Before:{10,50,10,60,60,60,60,40,10,10,10,10,80,3,3,3}

After:{10,50,10,60,40,10,80,3}

Q7. (20p) Periodic Strings: A character string is said to have period k if it can be formed by concatenating one or more repetitions of another string of length k. For example, the string "abcabcabcabc" has period 3, since it is formed by 4 repetitions of the string "abc". It also has periods 6 (two repetitions of "abcabc") and 12 (one repetition of "abcabcabcabc"). Write a program to read a character string and determine its smallest period.

Example:

Input String: HoHoHoHoHoHoHoHoHoHo

Output 2 is the smallest period. ("Ho")