CENG 104
Computer Programming 2
HOMEWORK 2

Due by 23:59pm, 20.03.2016

Task 1

In this task, you will implement the following requirements on a linked list. Let the nodes in the list have the following structure:

```
struct node
{
  int data;
  struct node* next;
};
```

Req. 1.0

Write the function `struct node* insertNode(struct node* head,int data)` that adds an element to the end of the list. The function should return the new head node to the list.

head, current head of the list

data, data to be inserted

return, updated head node

Req. 1.1

Write the function `void display(struct node* head)` that displays all the elements of the list.

head, pointer to the head node od the list

Req. 1.2

Write the function `struct node* find(struct node* head,int data)` that finds the elements that contains the given data, returns a pointer to the element in the list having the given data. The function should return NULL if the item does not exist.

head, pointer to the head node

data, given data to match

return, NULL if not found, pointer to the elemnt if found

Req. 1.3

Write the function `struct node* deleteNode(struct node* head,struct node* element)` that deletes the element pointed to by `element` (obtained using find). The function should return the updated head node. Make sure you consider the case when element points to the head node

head, pointer to the head node

element, pointer to the element to be removed

return, updated head node

Req. 1.4

Write a recursive function `int countNodes(node * head)` that calculates the number of nodes in a linked list.

****

Task 2

In this task, you will implement the given inorder, preorder and postorder tree traversal algorithms. Let the nodes in the tree have the following structure.

```
struct tnode
{
  int data;
  struct tnode* left;
  struct tnode* right;
};
```

Req. 2.0 Add Node

Write the function `void insert(int data)` to insert elements 3, 1, 0, 2, 8, 6, 5, 9 in the same order.

Req. 2.1 Preorder Traversal

```
Algorithm Preorder($x$)
Input: $x$ is the root of a subtree.
1.  if $x \neq$ NULL
2.     then output key($x$);
3.          Preorder(left($x$));
4.          Preorder(right($x$));
```

Req. 2.2 Postorder Traversal

```
Algorithm Postorder($x$)
Input: $x$ is the root of a subtree.
1.  if $x \neq$ NULL
2.     then Postorder(left($x$));
3.          Postorder(right($x$));
4.          output key($x$);
```

Req. 2.3 Inorder Traversal

```
Algorithm Inorder($x$)
Input: $x$ is the root of a subtree.
1.  if $x \neq$ NULL
2.     then Inorder(left($x$));
3.          output key($x$);
4.          Inorder(right($x$));
```

IMPORTANT:

- Make sure that your files are named in correct format.
- Please make a good design. The code should be properly written such that unnecessary information should be prevented.
- Use pre-defined methods if you need.
- Try to use whatever you learned about C programming until now in your codes.
- All requirements should be met.
- This homework should be done individually.
- Please do it on your own, you can read tutorials and forums, many examples exist on the internet.
- You should be sure to understand whatever you write in your code, we will randomly select and ask you to describe your code.
- For your questions, use mailgroup. Ask to everybody so that everybody sees the answers. No private mails for the homework please.
- Cheating is forbidden, no excuse will be accepted.