Dylan Buchheim

Group N

Lab 502 / Lecture 004

Questions:

I had no questions.

Acknowledgement:

I was the only one who contributed to my code.

//Main Pseudocode:

Data: Node is a struct type

name, stores the name of the person for that node

next, a pointer that points to the next node in the list

head, a pointer that points to the first node in the list

tail, a pointer that points to the last node in the list

Plan: //P1: initialize the head node with malloc(), and set the name to “” and next to NULL.

//P2: initialize the tail node equal to the head node.

//P3: call the MenuSelect function to bring up the users options.

//Function Definition:

Function Name: MenuSelect

Input: N.A.

Output: N.A.

Side Effect: prints the menu of users options and calls the correct function depending on the users menu selection.

Data: selection, a char which stores the users selection from the menu.

firstName[], an array of chars which stores the first name of the person who is removed from the front of the list.

name[], an array of chars which stores the users input name when they want to add someone to the end of the list.

Plan: //P1: print out the menu options.

//P2: get user input for selection.

//P3: open a switch statement based on selection.

//P3.1 case ‘0’ call a customer.

//P3.1.1: if the head->name is equal to “”, print that the line is empty.

//P3.1.2: otherwise deleteFirstName(firstName) and print firstName.

//P3.1.3: call MenuSelect to keep using the menu.

//P3.2: case ‘1’ add a customer.

//P3.2.1: get user input for name.

//P3.2.2: insert the name into the list, insert(name).

//P3.2.3: call MenuSelect to keep using the menu.

//P3.3: case ‘2’ exit.

//P3.3.1: exit the program.

//P3.4: default case.

//P3.4.1: print that the input was invalid and restart the menu, MenuSelect().

//Function Definition:

Function Name: DeleteFirstName

Input: \*name, a pointer which points to the passed in array of characters

Output: N.A.

Side Effect: removes the first node from the list and moves the head node forward.

Data: \*tempNode, a pointer for a temporary node used to release the memory used by the deleted node.

Plan: //P1: copy the head->name into name.

//P2: set tempNode = head.

//P3: set head = head->next.

//P4: release the memory held by tempNode, free(tempNode).

//Function Definition:

Function Name: Insert

Input: \*name, a pointer which points to the passed name that will be inserted.

Output: N.A.

Side Effect: a new node is created with the new name and added to the end of the list and the tail node is updated to reflect the new last node.

Data: current, a node pointer which points to the current tail node.

Plan: //P1: if the current node is the head node.

//P1.1: copy name into head->name

//P1.2: initialize head->next to a new node, malloc(sizeof(struct Node)).

//P1.3: update the tail node to reflect the new end node, tail = head->next.

//P2: if the current node is not the head node.

//P2.1: copy name into current->name.

//P2.2: initialize current->next to a new node, malloc(sizeof(struct Node)).

//P2.3: update the tail node to reflect the new end node, tail = current->next.

//P2.4: set the tail node name to “” and next to NULL.