**Data Structures and Algorithms**

**Lab 3**

**Submitted To:**

Mr. Dilshad Sabir

**Submitted By:**

Manaal Waseem

FA18-BCE-074

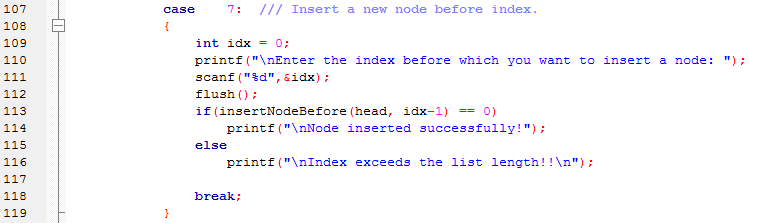
**In Lab:**

**Task 1:**

**‘Inserting nodes at the end’ and ‘inserting node after a given node’ are already implemented in ‘SinglyLinkedList.c’. Your task is to implement ‘insert at the beginning’ and ‘insert before’ functions in the file ‘SinglyLinkedList.c’.**

**Program:** In this program, already provided code has been compiled and its working has been analyzed. The function ***“listLength”*** has been implemented by calling it in *‘main.c’* and passing first node **‘head’** as argument to the function. As a result; length of list is printed on the console.

**‘Insert at the beginning’**

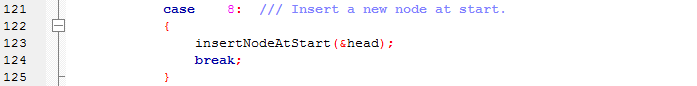


main.c



SinglyLinkedList.c

**‘Insert before’**

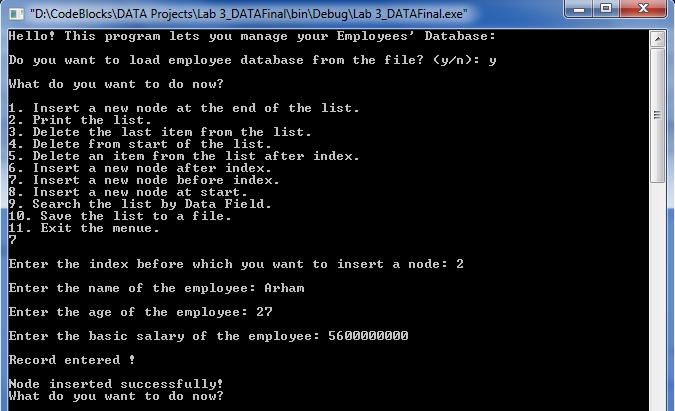


main.c

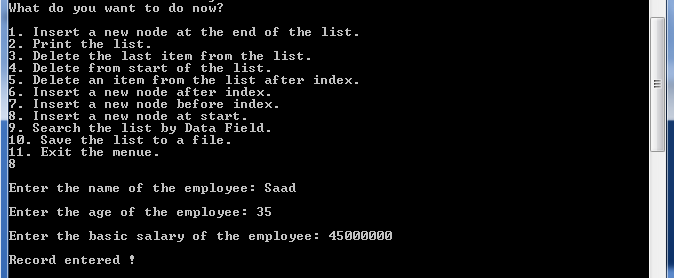


SinglyLinkedList.c

**Output:**



**‘Insert at the beginning’**



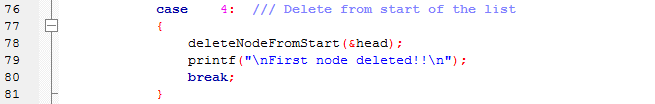
**‘Insert before’**

**Task 2:**

**Deleting a node from the end is already implemented in ‘SinglyLinkedList.c’ your task is to implement ‘delete from beginning’ and ‘delete after’ a given node.**

**Program:** In this program, a function ***“searchNode”*** takes first node of the linked list and the reference age as its argument. Variables **‘i’** and **‘flag’** are initialized as **1**and **false** respectively. A pointer *‘current’* of type **struct employee** is initialized with the first node of list. Function ***“isEmpty”*** checks if the list is empty otherwise while loop iterates till **‘next’** pointer of *‘current’* doesn’t point to **NULL** and checks if **‘age’** of *‘current’* is equal to *‘key’* i.e. reference age entered by user; **flag** is marked **‘true’** and position of respective node is printed on the console depending upon variable **‘i’**. If **flag** remains 0, a message of required node not found is displayed on the console.

**‘Delete from beginning’**

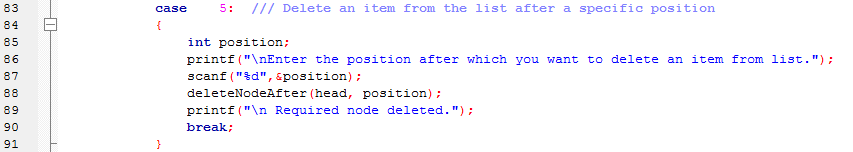


main.c

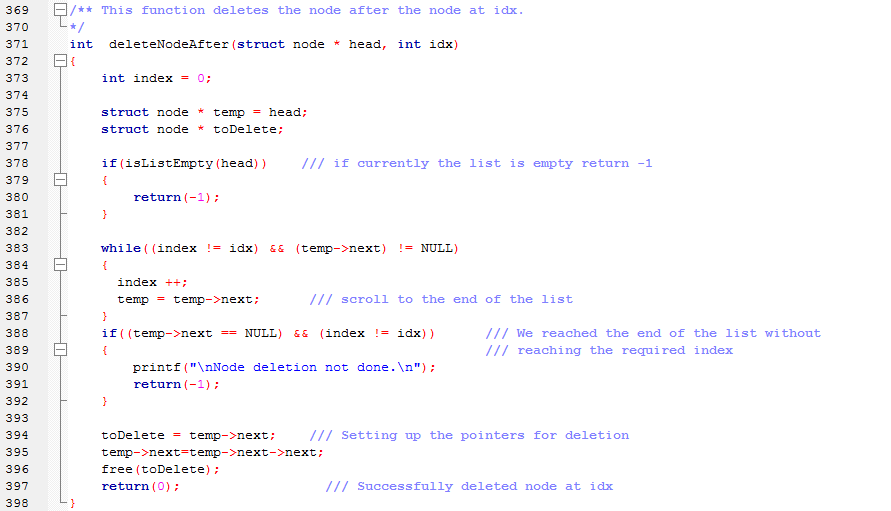


SinglyLinkedList.c

**‘Delete after’**



main.c



SinglyLinkedList.c

**Output:**

****

**‘Delete from beginning’**



**‘Delete after’**

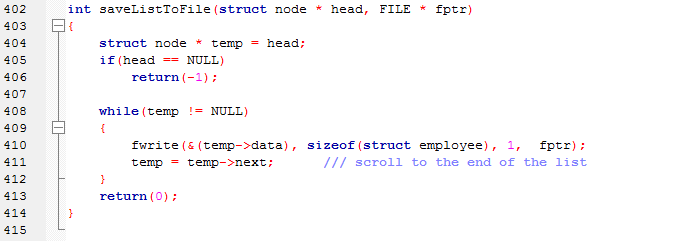
**Post Lab:**

**Delete the last node of the linked list.**

**Program:** In this program, a function ***“deleteLastNode”*** takes first node of the linked list as its argument. Function ***“isEmpty”*** checks if the list is empty otherwise pointers *‘toDelete’* and *‘secondLastNode’* are initialized as *‘first’*. While loop then iterates till **‘next’** pointer of *‘toDelete’* doesn’t point to **NULL** i.e. until second last node is found. If the node to be deleted is the first node then *‘t*o*Delete’* is equal to *‘first’* thus *‘*first’ is made to point **NULL** otherwise *‘secondLastNode’* is made to point **NULL** and memory location of *‘toDelete’* is freed using **‘free( )’**. Finally, the message of successful deletion is printed on the console.



main.c



SinglyLinkedList.c

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THE END**