

# Data Structures and Algorithms

## Lab Report

### Lab03



Group Members Name & Reg #:	<b><u>Muhammad Haris Irfan</u></b> <b>(FA18-BCE-090)</b>
Class	Data Structures and Algorithms CSC211 ( <b>BCE-3B</b> )
Instructor's Name	Dilshad Sabir

# In Lab Tasks

## Question no: 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*’.

## Solution:

The code of the following code is attached below:

### INSERT AT THE BEGINNING

```
int insertNodeAtStart(struct node ** head)
{
    /** Complete this function **/
    if(head==NULL)
    {
        *head= (struct node *) malloc(sizeof(struct node));

    }
    struct node * new_node = (struct node *) malloc(sizeof(struct node));

    inputNodeData(new_node);

    new_node->next=*head;
    *head=new_node;
}
```

the result is attached below,

```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
8. Insert a Node at Start.
9. Delete a node from start.
10. Delete a node after a node.
11. Exit the menu.
8
Inserting a Node at Start.

Enter the name of the employee: Haris
Enter the age of the employee: 21
Enter the basic salary of the employee: 1212121
Record entered !

What do you want to do now?
1. Insert a new node at the end of the list.
2. Print the list.
3. Delete the last item from the list.
4. Insert a new node after index.
5. Search the list by Data Field.
6. Save list to file.
7. Insert a Node before a Node.
8. Insert a Node at Start.
9. Delete a node from start.
10. Delete a node after a node.
11. Exit the menu.
```

```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
2
Start of list:
Name: Haris
Age: 21
Basic Salary: 1212121.000000
Name: Junaid
Age: 43
Basic Salary: 220000.000000
Name: Haris
Age: 21
Basic Salary: 2222222.000000
Name: Moazzam
Age: 43
Basic Salary: 225000.000000
Name: Ikram
Age: 36
Basic Salary: 230000.000000
Name: Zaheer
Age: 28
Basic Salary: 76000.000000
Name: haris
Age: 21
```

## INSERT BEFORE A NODE

The code of the following code is attached below:

```
int insertNodeBefore(struct node * head, int idx)
{
    /** Complete this function **/
    int index = 0;
    struct node * temp = head;

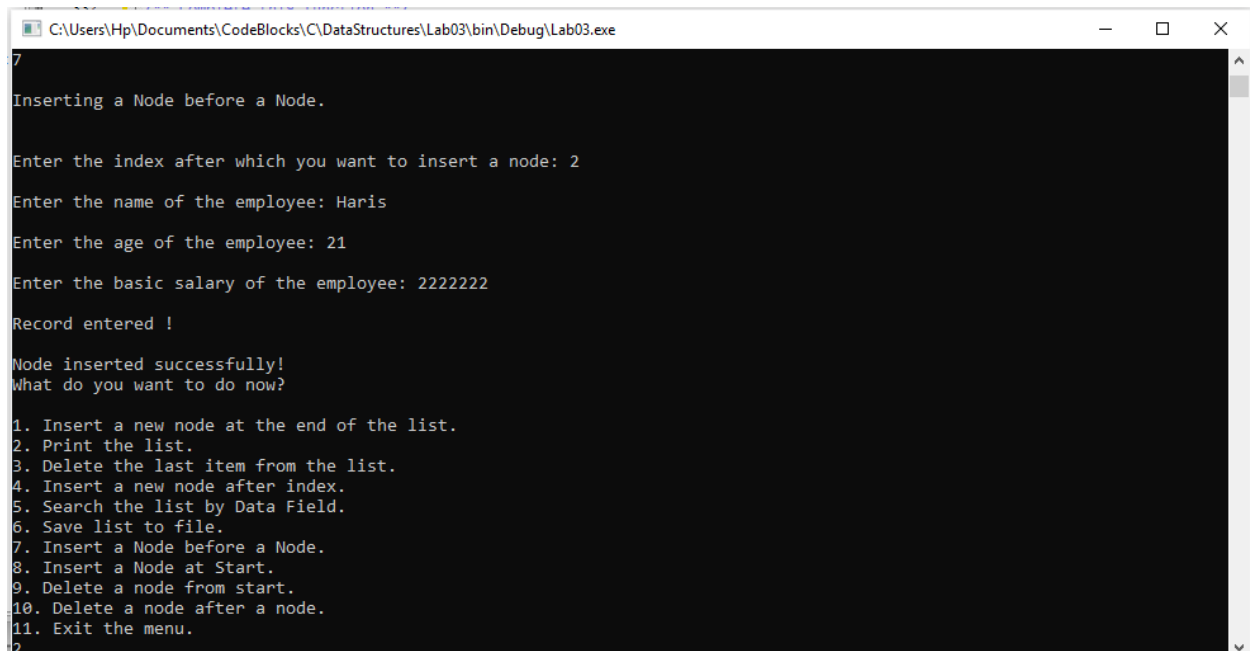
    if(isListEmpty(head))    /// if currently the list is empty return -1
    {
        return(-1);
    }
    ///create a new node
    struct node * new_node = (struct node *) malloc(sizeof(struct node));

    inputNodeData(new_node);    /// get data for the newly created node from the user.

    while((index != (idx-1) && (temp->next) != NULL))
    {
        index++;
        temp = temp->next;    /// scroll to the end of the list
    }
    if((temp->next == NULL) && (index != (idx-1)) )    /// We reached the end of the list without
    {    /// reaching the required index
        free(new_node);
        printf("\nNode insertion not done. New data is discarded!!\n");
        return(-1);
    }

    new_node->next = temp->next;    /// Setting up the pointers for insertion
    temp->next = new_node;
    return(0);    ///
}
```

the result is attached below,



```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
7
Inserting a Node before a Node.

Enter the index after which you want to insert a node: 2

Enter the name of the employee: Haris

Enter the age of the employee: 21

Enter the basic salary of the employee: 2222222

Record entered !

Node inserted successfully!
What do you want to do now?

1. Insert a new node at the end of the list.
2. Print the list.
3. Delete the last item from the list.
4. Insert a new node after index.
5. Search the list by Data Field.
6. Save list to file.
7. Insert a Node before a Node.
8. Insert a Node at Start.
9. Delete a node from start.
10. Delete a node after a node.
11. Exit the menu.
2
```

```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
Name: Omar
Age: 38
Basic Salary: 175000.000000

Name: Junaid
Age: 43
Basic Salary: 220000.000000

Name: Haris
Age: 21
Basic Salary: 2222222.000000

Name: Moazzam
Age: 43
Basic Salary: 225000.000000

Name: Haris
Age: 21
Basic Salary: 12345667.000000

Name: Ikram
Age: 36
Basic Salary: 230000.000000

Name: Zaheer
Age: 28
Basic Salary: 76000.000000

Name: haris
Age: 21
```

=====

## Question no: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.

## Solution

The code is shown below,

### DELETING FROM THE BEGINING

```
void deleteNodeFromStart(struct node ** head)
{
    if(*head==NULL)
    {
        printf("\nthe list is already empty\n");
    }
    else
    {
        if ((*head)->next==NULL)
        {
            free(*head);
        }
        else
        {
            *head=(*head)->next;
        }
    }
    printf("\nThe Node from the start is Deleted\n");
}
```

The Result of the following code is attached below:

```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
11. Exit the menu.
9
Deleting a node from start

The Node from the start is Deleted

What do you want to do now?
1. Insert a new node at the end of the list.
2. Print the list.
3. Delete the last item from the list.
4. Insert a new node after index.
5. Search the list by Data Field.
6. Save list to file.
7. Insert a Node before a Node.
8. Insert a Node at Start.
9. Delete a node from start.
10. Delete a node after a node.
11. Exit the menu.
```

```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
2
Start of list:
Name: Junaid
Age: 43
Basic Salary: 220000.000000
Name: Haris
Age: 21
Basic Salary: 2222222.000000
Name: Moazzam
Age: 43
Basic Salary: 225000.000000
Name: Haris
Age: 21
Basic Salary: 12345667.000000
Name: Ikram
Age: 36
Basic Salary: 230000.000000
Name: Zaheer
Age: 28
Basic Salary: 76000.000000
```

## DELETING AFTER A NODE

```
int deleteNodeAfter(struct node * head, int idx)
{
    /** Complete this function **/

    int index=0;
    struct node * movinghead = head;
    struct node * del= head;
    struct node * last =head;

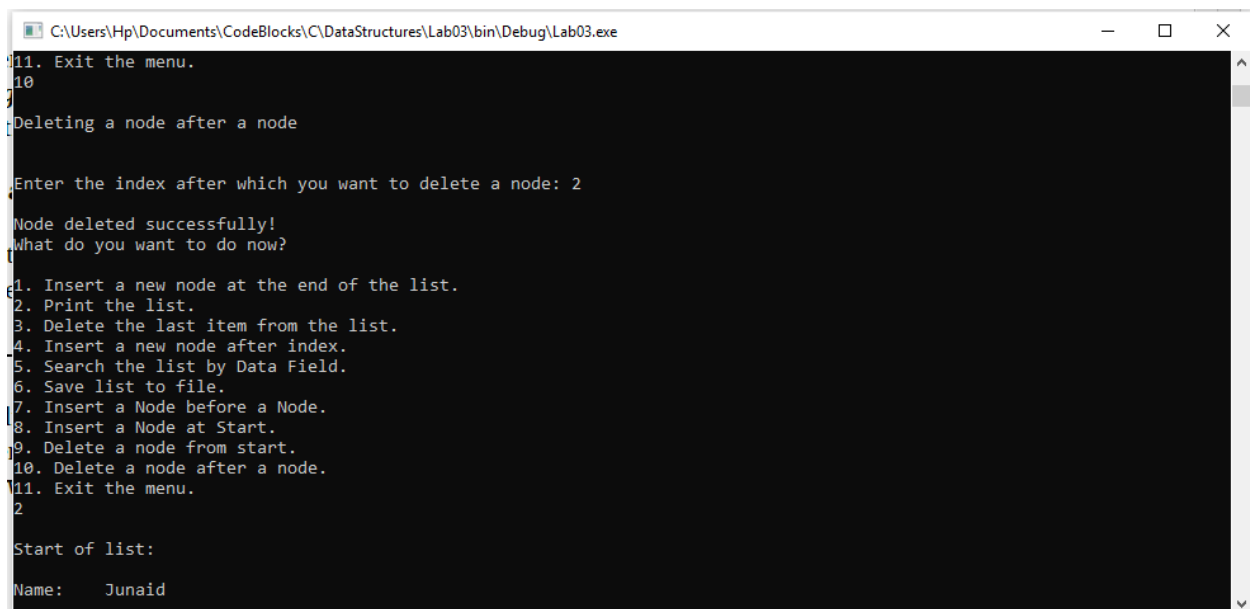
    if(head==NULL)
    {
        printf("\n The List is Empty\n");
        return;
    }

    for(index=1;index<=idx;index++)
    {
        del=del->next;
        last=last->next;
    }
    last=last->next;
    del->next=last->next;

    free(last);

    return(0);
}
```

the Result of the following code is attached below:



```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
11. Exit the menu.
10
Deleting a node after a node

Enter the index after which you want to delete a node: 2

Node deleted successfully!
What do you want to do now?

1. Insert a new node at the end of the list.
2. Print the list.
3. Delete the last item from the list.
4. Insert a new node after index.
5. Search the list by Data Field.
6. Save list to file.
7. Insert a Node before a Node.
8. Insert a Node at Start.
9. Delete a node from start.
10. Delete a node after a node.
11. Exit the menu.
2

Start of list:
Name: Junaid
```



```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
Start of list:
Name:   Junaid
Age:    43
Basic Salary:  220000.000000

Name:   Haris
Age:    21
Basic Salary:  2222222.000000

Name:   Moazzam
Age:    43
Basic Salary:  225000.000000

Name:   Ikram
Age:    36
Basic Salary:  230000.000000

Name:   Zaheer
Age:    28
Basic Salary:  76000.000000

Name:   haris
Age:    21
Basic Salary:  12312312.000000

End of list.
```

=====

## POST LAB

### Question no:3

Reading database from a file on the hard disk is already implemented. Your first task is to study and understand this implementation. Then you will have to implement the write to file function '***saveListToFile()***'. Submit a report on your implementation.

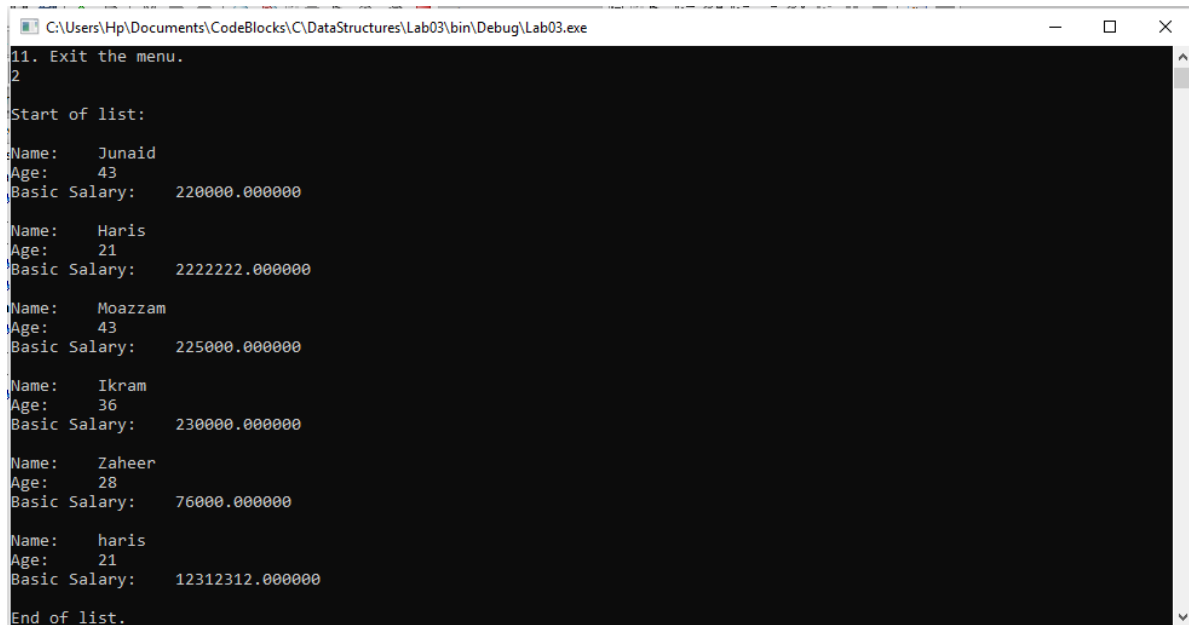
### Solution

The code is shown below for the given program and its results are given below,

```
int saveListToFile(struct node * head, FILE * fptr)
{
    struct node * temp = head;
    if(head == NULL)
        return(-1);

    while(temp != NULL)
    {
        fwrite(&(temp->data), sizeof(struct employee), 1, fptr);
        temp = temp->next;    /// scroll to the end of the list
    }
    return(0);
}
```

The Result of the following code is attached below:



```
C:\Users\Hp\Documents\CodeBlocks\C\DataStructures\Lab03\bin\Debug\Lab03.exe
11. Exit the menu.
2
Start of list:
Name: Junaid
Age: 43
Basic Salary: 220000.000000
Name: Haris
Age: 21
Basic Salary: 2222222.000000
Name: Moazzam
Age: 43
Basic Salary: 225000.000000
Name: Ikram
Age: 36
Basic Salary: 230000.000000
Name: Zaheer
Age: 28
Basic Salary: 76000.000000
Name: haris
Age: 21
Basic Salary: 12312312.000000
End of list.
```

THE END