

**Instructions: Please read carefully**

- Please rename this file as only your ID number (e.g. 18-\*\*\*\*\*-1.doc or 18-\*\*\*\*\*-1.pdf).
- Submit the file before **11:59pm on 23/03/2020** in the Portal Lab Performance section labeled **Lab task 9**. If you cannot complete the full task, do not worry. Just upload what you have completed.

**Name:- Amit Podder**

**ID:- 20-42273-1**

**Section:- [ F ]**

**1. Do the following to write program for a Single Linked List:**

1. Create a singly linked list by **inserting** node one by one at the end.
2. **Display** your list
3. **Search** an item into your linked list.
4. **Insert** a new item at a specific position (at the beginning and after a given node)
5. **Delete** an item from the list (at beginning, at last and at middle)

Take users input from the console to perform the operations.

**Your code here:-**

```
#include<iostream>

using namespace std;

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

class linkedlist
{
public:
    struct node *head;
    linkedlist()
    {
        head=NULL;
    }

    struct node* getHead()
    {
        return head;
    }

    //push in end
    void pushEnd(int a)
    {
        struct node *newnode= new node;
        newnode->data=a;
        newnode->next=NULL;
        if(head==NULL)
        {
            head=newnode;
```

```

    }
    else
    {
        struct node *temp= new node;
        temp=head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newnode;
    }
}

```

//push element at the front

void pushFront(int a)

```

{
    struct node *temp= new node;
    temp->data=a;
    temp->next=head;
    head=temp;
}

```

//push after a specific location

void pushMid(int a , int b)

```

{
    struct node *temp=new node;
    struct node *newnode=new node;
    newnode->data=a;
    temp=head;
    for(int i=1 ; i<b ;i++)
    {
        temp=temp->next;
    }
    newnode->next=temp->next;
    temp->next=newnode;
}

```

//calculate how many elements are in list

int count(){

```

    int count=0;
    struct node *temp= new node;
    temp=head;
    while(temp!=NULL)
    {
        temp=temp->next;
        count++;
    }
    return count;
}

```

//delete from begeing

void deleteFront()

```

{

```

```

        if(head==NULL)
        {
            cout<<"List is empty"<<endl;
        }
        else
        {
            struct node *temp= new node;
            temp=head;
            head=head->next;
            delete temp;
        }
    }

//delete from end
void deleteEnd()
{
    if(head==NULL)
    {
        cout<<"List is empty"<<endl;
    }
    else
    {
        struct node *temp= new node;
        struct node *temprev= new node;
        temp=head;
        while(temp->next!=NULL)
        {
            temprev=temp;
            temp=temp->next;
        }
        if(temp==head)
        {
            head=NULL;
        }
        else
        {
            temprev->next=NULL;
        }
        delete temp;
    }
}

//delet from mid
void deleteMid(int b)
{
    if(head==NULL)
    {
        cout<<"List is empty"<<endl;
    }
    else
    {
        struct node *temp= new node;

```

```

        struct node *tempnext= new node;
        temp=head;
        for(int i =1 ; i<b-1 ; i++){
            temp=temp->next;
        }
        tempnext=temp->next;
        temp->next=tempnext->next;
        delete tempnext ;
    }
}
void show ()
{
    if(head==NULL)
    {
        cout<<"List is Empty\n";
    }
    else
    {
        cout<<"Values are: ";
        struct node *temp= new node;
        temp=head;
        while(temp!=NULL)
        {
            cout<<temp->data<<" ";
            temp=temp->next;
        }
    }
    cout<<endl;
}
bool serch(int a)
{
    struct node *temp= new node;
    temp=head;
    while(temp!=NULL)
    {
        if(temp->data==a)
        {
            return true;
        }
        temp=temp->next;
    }
    return false ;
}
};

int main ()
{
    linkedlist l1;
    l1.pushEnd(10);
    l1.pushEnd(20);
    l1.pushEnd(30);
    l1.pushEnd(40);
    bool flag=true;

```

```

while(flag)
{
    cout<<"____Single Linked List____"<<endl;
    cout<<"\nWhich Operation do you want to do? \n1. Insert\n2. Delete\n3. Show\n4. Search"<<endl;
    int b;
    cin>>b;

    //insert
    if(b==1)
    {
        cout<<"Where do you want to insert ?\n1. At the begening\n2. At the end\n3. After a specific
index.\n";

        int c;
        cin>>c;
        if(c==1)
        {
            cout<<"Enter a value: "<<endl;
            int a;
            cin>>a;
            l1.pushFront(a);
            cout<<"Inserted"<<endl;
        }
        else if(c==2)
        {
            cout<<"Enter a value: "<<endl;
            int a;
            cin>>a;
            l1.pushEnd(a);
            cout<<"Inserted"<<endl;
        }
        else if(c==3)
        {
            cout<<"Enter index :\n";
            int b;
            cin>>b;
            cout<<"Enter a value: "<<endl;
            int a;
            cin>>a;
            if(b<l1.count())
            {
                l1.pushMid(a,b);
                cout<<"Inserted"<<endl;
            }
            else
                cout<<"Can not be inserted. Enter a valid index \n";
        }
    }

    //delete
    else if(b==2)
    {
        cout<<"Where do you want to do delete operation ?\n1. At the begening\n2. At the end\n3. On a
specific index.\n";
    }
}

```

```

        int c;
        cin>>c;
        if(c==1)
        {
            l1.deleteFront();
            cout<<"Element deleted\n";
        }
        else if(c==2)
        {
            l1.deleteEnd();
            cout<<"Element deleted\n";
        }
        else if(c==3)
        {
            int b;
            cout<<"Enter deleting index: ";
            cin>>b;
            l1.deleteMid(b);
        }
    }

    //show
    else if(b==3)
    {
        l1.show();
    }

    //search
    else if(b==4)
    {
        cout<<"Enter element to serch: ";
        int a; cin>>a;
        if(l1.serch(a))
        {
            cout<<"Element Found"<<endl;
        }
        else
        {
            cout<<"Element not found\n";
        }
    }
    else
    {
        flag=false;
    }
}
return 0;
}

```

**Your whole Screenshot here: (Console Output):-**

C:\Users\USER\Desktop\1\main.exe

```
-- _ _ _Single Linked List_ _ _ _
Which Operation do you want to do?
1. Insert
2. Delete
3. Show
4. Search
3
Values are: 10 20 30 40
-- _ _ _Single Linked List_ _ _ _
Which Operation do you want to do?
1. Insert
2. Delete
3. Show
4. Search
1
Where do you want to insert ?
1. At the begening
2. At the end
3. After a specific index.
1
Enter a value:
50
Inserted
-- _ _ _Single Linked List_ _ _ _
Which Operation do you want to do?
1. Insert
2. Delete
```

C:\Users\USER\Desktop\1\main.exe

```
Which Operation do you want to do?
1. Insert
2. Delete
3. Show
4. Search
3
Values are: 50 10 20 30 40
-- _ _ _Single Linked List_ _ _ _
Which Operation do you want to do?
1. Insert
2. Delete
3. Show
4. Search
1
Where do you want to insert ?
1. At the begening
2. At the end
3. After a specific index.
2
Enter a value:
60
Inserted
-- _ _ _Single Linked List_ _ _ _
Which Operation do you want to do?
1. Insert
2. Delete
3. Show
4. Search
```

C:\Users\USER\Desktop\1\main.exe

Which Operation do you want to do?

1. Insert
2. Delete
3. Show
4. Search

3

Values are: 50 10 20 30 40 60

\_ \_ \_ \_Single Linked List\_ \_ \_ \_

Which Operation do you want to do?

1. Insert
2. Delete
3. Show
4. Search

2

Where do you want to do delete operation ?

1. At the begening
2. At the end
3. On a specific index.

1

Element deleted

\_ \_ \_ \_Single Linked List\_ \_ \_ \_

Which Operation do you want to do?

1. Insert
2. Delete
3. Show
4. Search

3

C:\Users\USER\Desktop\1\main.exe

Element deleted

\_ \_ \_ \_Single Linked List\_ \_ \_ \_

Which Operation do you want to do?

1. Insert
2. Delete
3. Show
4. Search

3

Values are: 10 20 30 40 60

\_ \_ \_ \_Single Linked List\_ \_ \_ \_

Which Operation do you want to do?

1. Insert
2. Delete
3. Show
4. Search

2

Where do you want to do delete operation ?

1. At the begening
2. At the end
3. On a specific index.

2

Element deleted

\_ \_ \_ \_Single Linked List\_ \_ \_ \_

Which Operation do you want to do?

1. Insert
2. Delete
3. Show



```
C:\Users\USER\Desktop\1\main.exe
Where do you want to do delete operation ?
1. At the begening
2. At the end
3. On a specific index.
2
Element deleted
_ _ _ _Single Linked List_ _ _ _

Which Operation do you want to do?
1. Insert
2. Delete
3. Show
4. Search
3
Values are: 10 20 30 40
_ _ _ _Single Linked List_ _ _ _

Which Operation do you want to do?
1. Insert
2. Delete
3. Show
4. Search
5

Process returned 0 (0x0)   execution time : 119.636 s
Press any key to continue.
```

## 2. Solve and submit at least one of the following problems.

1. Write a code to implement Doubly Linked List operations
2. Implement Stack using Linked List
3. Implement Queue using Linked List
4. Implement a program to sort the elements in a Linked List

Your code here:

Your whole Screenshot here: (Console Output):