**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-112L Data Structure** **& Algorithm Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 4th**

**Submission Date: 26th April 2022**

**Submitted to: Engr. Rafi-Ullah**

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

**Instructor Signature**

**Lab No. 06**

**Single link list, Double link list, Link list operations**

**Objectives:**

In this lab we will be discussing about Single link list, Double link list, Link list operations in detail. This is one of the most important concepts in C++ language.

The Lab objective is:

* To understand Single and double link lists.
* To Implement link lists in C++

**Tools/Software Required:**

* All the tasks are implemented on DEV C++.

**Introduction:**

A linked list is just a chain of nodes, with each subsequent node being a child of the previous one. Many programs rely on linked lists for their storage because these don't have any evident restrictions. For example, the array list we did earlier could not grow or shrink, but node-based ones can! This means there is no limit (other than the amount of memory) on the number of elements they can store.

**Singly Linked Lists Basics:**

* A singly linked list is a concrete data structure consisting of a sequence of nodes
* It has a head (or first) node pointer indicating the first node in list
* It could have optionally a tail pointer node indication the last node in list
* Each node stores
* Element (data)
* Link to the next node
* The null object (at the end) is denoted as.

The elements of a linked list are called the nodes. A node has two fields i.e. data and next. The data field contains the data being stored in that specific node. It cannot just be a single variable. There may be many variables presenting the data section of a node. The next field contains the address of the next node. So this is the place where the link between nodes is established.

**Doubly Linked Lists**

The doubly linked list allows us to go in both directions in a linked list:

* Forward
* Reverse
* A variety of quick update operations, including insertion and removal at both ends, and in the middle.
* A node in a doubly linked list stores two references A next link, which points to the next node in the list.
* A prev link, which points to the previous node in the list.

**Lab Tasks:**

**Task 1:**

Write and test a method public int size () to count the number of nodes in the linked list.

**Code:**

**#include <iostream>**

**using namespace std;**

**class node**

**{**

**private:**

**int data;**

**node\* next;**

**node \*head=NULL;**

**node \*ptr;**

**public:**

**void InsertNode(int value)**

**{**

**node\* temp=new node();**

**temp->data=value;**

**temp->next=NULL;**

**if(head==NULL)**

**{**

**head=temp;**

**ptr=head;**

**}**

**else**

**{**

**ptr->next=temp;**

**ptr=ptr->next;**

**}**

**}**

**void show()**

**{**

**ptr=head;**

**cout<<"\n\nThe values in the linked list are :\n";**

**while(ptr->next!=NULL)**

**{**

**cout<<ptr->data<<"\t";**

**ptr=ptr->next;**

**}**

**cout<<ptr->data;**

**}**

**int size()**

**{**

**int size=0;**

**ptr=head;**

**cout<<endl;**

**while(ptr->next!=NULL)**

**{**

**ptr=ptr->next;**

**size++;**

**}**

**size++;**

**return size;**

**}**

**};**

**int main()**

**{**

**int value,num,i=1;**

**node n;**

**do**

**{**

**cout<<"Enter number "<<i<<" (-1 to exit) : ";**

**cin>>value;**

**if(value!=-1)**

**n.InsertNode(value);**

**i++;**

**}**

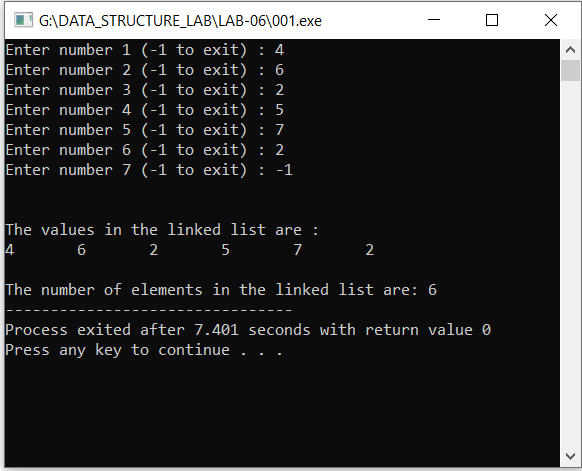
**while(value!=-1);**

**n.show();**

**cout<<"\nThe number of elements in the linked list are: "<<n.size();**

**}**

**Output:**

****

**Task # 02:**

Write and test a method public Boolean search (int n) to find out whether the given data exists or not in the linked list.

**Code:**

**#include <iostream>**

**using namespace std;**

**class node**

**{**

**private:**

**int data;**

**node\* next;**

**node \*head=NULL;**

**node \*ptr;**

**public:**

**void InsertNode(int value)**

**{**

**node\* temp=new node();**

**temp->data=value;**

**temp->next=NULL;**

**if(head==NULL)**

**{**

**head=temp;**

**ptr=head;**

**}**

**else**

**{**

**ptr->next=temp;**

**ptr=ptr->next;**

**}**

**}**

**void show()**

**{**

**ptr=head;**

**cout<<"\n\nThe values in the linked list are :\n";**

**while(ptr->next!=NULL)**

**{**

**cout<<ptr->data<<"\t";**

**ptr=ptr->next;**

**}**

**cout<<ptr->data;**

**}**

**bool find(int find)**

**{**

**int found;**

**ptr=head;**

**while(ptr->next!=NULL)**

**{**

**if(ptr->data == find)**

**return true;**

**ptr=ptr->next;**

**}**

**if(ptr->data == find)**

**return true;**

**return false;**

**}**

**};**

**int main()**

**{**

**int value,search,i=1;**

**node n;**

**do**

**{**

**cout<<"Enter number "<<i<<" (-1 to exit) : ";**

**cin>>value;**

**if(value!=-1)**

**n.InsertNode(value);**

**i++;**

**}**

**while(value!=-1);**

**n.show();**

**cout<<"\n\nEnter the number to find : ";**

**cin>>search;**

**bool flag=n.find(search);**

**if(flag)**

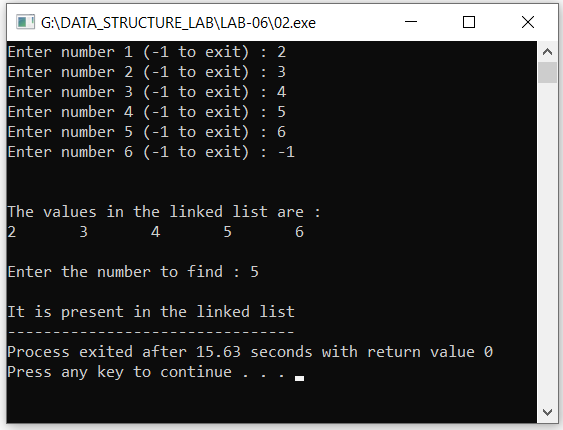
**cout<<"\nIt is present in the linked list";**

**else**

**cout<<"\nIt is NOT present in the linked list";**

**}**

**Output:**

****

**Task # 03:**

Swap nodes in a linked list without swapping data.

**Code:**

**#include <iostream>**

**using namespace std;**

**class node**

**{**

**private:**

**int data;**

**node\* next;**

**node \*head=NULL;**

**node \*ptr;**

**public:**

**void InsertNode(int value)**

**{**

**node\* temp=new node();**

**temp->data=value;**

**temp->next=NULL;**

**if(head==NULL)**

**{**

**head=temp;**

**ptr=head;**

**}**

**else**

**{**

**ptr->next=temp;**

**ptr=ptr->next;**

**}**

**}**

**void show()**

**{**

**ptr=head;**

**cout<<"\n\nThe values in the linked list are :\n";**

**while(ptr->next!=NULL)**

**{**

**cout<<ptr->data<<"\t";**

**ptr=ptr->next;**

**}**

**cout<<ptr->data;**

**}**

**bool find(int find)**

**{**

**int found;**

**ptr=head;**

**while(ptr->next!=NULL)**

**{**

**if(ptr->data == find)**

**return true;**

**ptr=ptr->next;**

**}**

**if(ptr->data == find)**

**return true;**

**return false;**

**}**

**void swaping(int num1,int num2)**

**{**

**ptr=head;**

**while(ptr->data!=num1)**

**{ptr=ptr->next; }**

**node\*temp=head;**

**while(temp->data!=num2)**

**{ temp=temp->next; }**

**temp->data=num1;**

**ptr->data=num2;**

**}**

**};**

**int main()**

**{**

**int value,no,search,i=1;**

**int num1,num2;**

**node n;**

**do**

**{**

**cout<<"Enter number "<<i<<" (-1 to exit) : ";**

**cin>>value;**

**if(value!=-1)**

**n.InsertNode(value);**

**i++;**

**}**

**while(value!=-1);**

**n.show();**

**cout<<"\n\nEnter the first number to find : ";**

**cin>>num1;**

**bool flag1=n.find(num1);**

**if(flag1)**

**cout<<"\nFirst number present in the linked list";**

**else**

**cout<<"\nFirst number is NOT present in the linked list";**

**cout<<"\n\nEnter the second number to find : ";**

**cin>>num2;**

**bool flag2=n.find(num2);**

**if(flag2)**

**cout<<"\nSecond number present in the linked list";**

**else**

**cout<<"\nSecond number is NOT present in the linked list";**

**if(flag1 && flag2)**

**{**

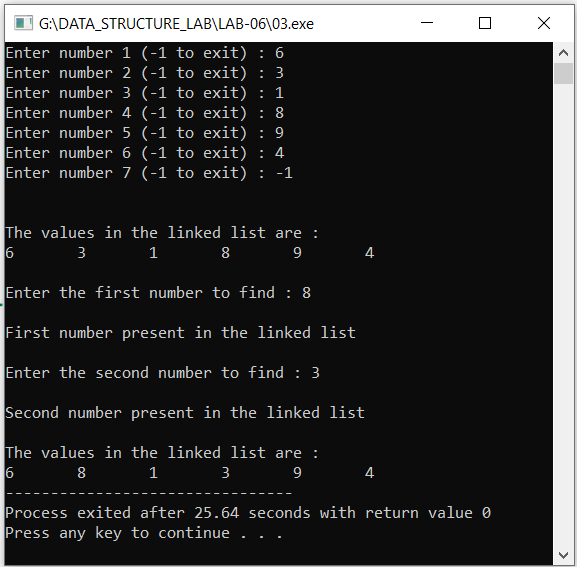
**n.swaping(num1,num2);**

**}**

**n.show();**

**}**

**Output:**

****

**Results & Observations:**

In this Lab I've learned about the concept of Single link list, Double link list, Link list operations & also understand that how we can make a node and then linked in the sequence. Further I understand the concept of Singly Linked List in which there’s only next pointer that points to the next node it means it’s one way method. On the other hand, in Double linked list there’re two pointers one links the node forward and other backward. In the first task, I've used class node in which I've used insert function to insert new node and create count variable that counts the number of the node. In the second task, I've used Bool function to check either the required data is available in the linked list or not. In the last task, I've used searching the two numbers in the linked list and if they are available then swapped them by using swap function.