**Bahria University, Lahore Campus**

Department of Computer Science

Lab Journal 06

**(Spring 2023)**

|  |  |  |
| --- | --- | --- |
| Course: | **Data Structures and Algorithm - Lab** | Date: \_27-04-2023\_ |
| Course Code: | CSL-221 | Max Marks: 10 |
| Faculty’s Name: | Fatima Zulfiqar |  |

Name: AFFAN AHMAD\_\_\_ Enroll No: \_03-134221-003\_\_\_ Class: \_BScs 3A\_\_

Objective(s):

Upon completion of this lab session, learners will be able to:

* Implement Singly Linked List and its helping functions (insertion from start, insert from end, insert from any point, delete from start, delete from end, delete from any point, and display)
* Count number of nodes present in a Linked List.

## Lab Tasks:

**Task 1**

Implement doubly linked list data structure and perform following operations.

* Insert from start ()
* Insert from end ()
* Delete from start ()
* Delete from end ()
* Display ()

**Note:** The program should contain main-menu in such a way that the user can select either of the options until desires. Additionally the input to the node should be taken from the user.

#include<iostream>

#include <stdlib.h>

using namespace std;

int t =0;

struct node

{

int data;

node \*next;

node \*head, \*tail;

node \*pre;

};

node \*head = NULL, \*tail = NULL;

void insert\_start(int value)

{

node \*temp = new node;

temp->data = value;

temp->next = NULL;

if (head == NULL)

{

head = temp;

tail = temp;

}

else

{

temp->next = head;

head->pre=temp;

head = temp;

}

}

void insertend(int value)

{

node \*temp = new node;

temp->data = value;

tail->next = temp;

temp->pre=tail;

tail = temp;

}

void deletefirst()

{

node \*temp;

temp = head;

head = head->next;

head->pre=NULL;

delete temp;

}

void deletelast()

{

node \*cur;

node \*p;

cur=head;

while(cur->next!=NULL)

{

p=cur;

cur=cur->next;

}

tail=p;

p->next=NULL;

cur->pre=NULL;

delete cur;

}

void display()

{

node \*temp;

temp = head;

while (temp != NULL)

{

cout << temp->data << " ";

temp = temp->next;

}

}

int main()

{

int val, ch,pos;

cout << "press 1 for insert at start " << endl;

cout << "press 2 for insert at end " << endl;

cout << "press 4 for delete at start " << endl;

cout << "press 5 for delete at end " << endl;

cout << "press 7 for display :" << endl;

cout << "press 9 for Exit :" << endl;

do{

cout << "Enter your choise :";

cin >> ch;

if (ch == 1)

{

++t;

cout << "Enter the value :"; cin >> val;

//createnode(val);

insert\_start(val);

}

if (ch == 7)

{

cout << "your list is " << endl;

display();

cout << endl;

}

if (ch == 2)

{

++t;

cout << "Enter your value :"; cin >> val;

insertend(val);

}

if (ch == 4)

{

t=t-1;

deletefirst();

}

if (ch == 5)

{

t=t-1;

deletelast();

}

if (ch==8)

{

if (t<=0)

{

cout <<"list is empty "<< endl;

}

else

{

cout <<"the total nodes in linklist is :"<< t<< endl;

}

}

if (ch == 9)

{

cout << "exit" << endl;

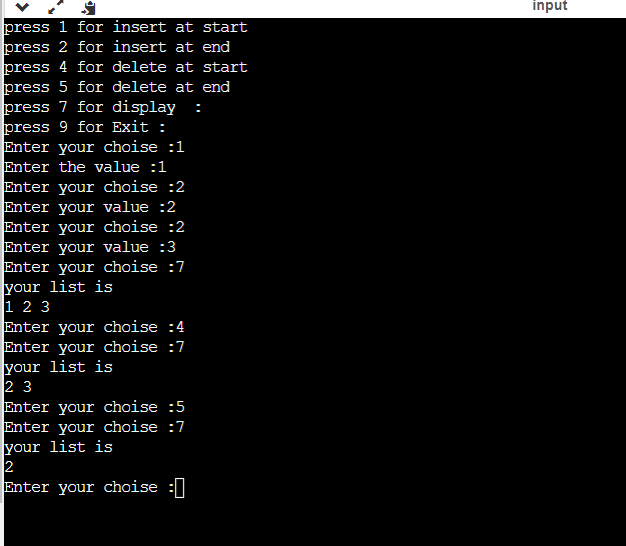
break;

}

} while (ch != 9);

return 0;

}



**Task 2**

Add additional function **remove\_duplicates ()** in Task 1. The function should first display all elements present in a linked list and then all duplicate values/numbers should be removed from the list. The final list should contain non-duplicate nodes and only one occurrence of duplicate number.

**Sample Output:**

Original Linked List: 2 3 1 4 5 6 2 3 7 1 2 5 4 4 3

After Removing duplicates: 2 3 1 4 5 6 7

#include<iostream>

#include <stdlib.h>

using namespace std;

int t =0;

struct node

{

int data;

node \*next;

node \*head, \*tail;

node \*pre;

};

node \*head = NULL, \*tail = NULL;

void insert\_start(int value)

{

node \*temp = new node;

temp->data = value;

temp->next = NULL;

if (head == NULL)

{

head = temp;

tail = temp;

}

else

{

temp->next = head;

head->pre=temp;

head = temp;

}

}

void insertend(int value)

{

node \*temp = new node;

temp->data = value;

tail->next = temp;

temp->pre=tail;

tail = temp;

}

void deletefirst()

{

node \*temp;

temp = head;

head = head->next;

head->pre=NULL;

delete temp;

}

void deletelast()

{

node \*cur;

node \*p;

cur=head;

while(cur->next!=NULL)

{

p=cur;

cur=cur->next;

}

tail=p;

p->next=NULL;

cur->pre=NULL;

delete cur;

}

void display()

{

node \*temp;

temp = head;

while (temp != NULL)

{

cout << temp->data << " ";

temp = temp->next;

}

}

void deletedublicate()

{

node \*cur;

node \*p;

node \*a;

node \*b;

int temp ;

a=head;

cur=head;

while(cur!=NULL)

{

p=cur->next;

while(p!=NULL)

{

if (cur->data>p->data)

{

temp=cur->data;

cur->data=p->data;

p->data=temp;

}

p=p->next;

}

cur=cur->next;

}

while(a->next!=NULL)

{

b=a;

a=a->next;

if (b->data!=a->data)

{

b->next=a;

a->pre=b;

}

for(int i=1;i<t;i++)

{

if(b->data==a->data)

{

a=a->next;

b->next=a;

a->pre=b;

}

}

}

}

int main()

{

int val, ch,pos;

cout << "press 1 for insert at start " << endl;

cout << "press 2 for insert at end " << endl;

cout << "press 3 for delete dublicate " << endl;

cout << "press 4 for delete at start " << endl;

cout << "press 5 for delete at end " << endl;

cout << "press 7 for display :" << endl;

cout << "press 9 for Exit :" << endl;

do{

cout << "Enter your choise :";

cin >> ch;

if (ch == 1)

{

++t;

cout << "Enter the value :"; cin >> val;

//createnode(val);

t++;

insert\_start(val);

}

if (ch==3)

{

deletedublicate();

}

if (ch == 7)

{

cout << "your list is " << endl;

display();

cout << endl;

}

if (ch == 2)

{

++t;

cout << "Enter your value :"; cin >> val;

insertend(val);

t++;

}

if (ch == 4)

{

t=t-1;

deletefirst();

}

if (ch == 5)

{

t=t-1;

deletelast();

}

if (ch==8)

{

if (t<=0)

{

cout <<"list is empty "<< endl;

}

else

{

cout <<"the total nodes in linklist is :"<< t<< endl;

}

}

if (ch == 9)

{

cout << "exit" << endl;

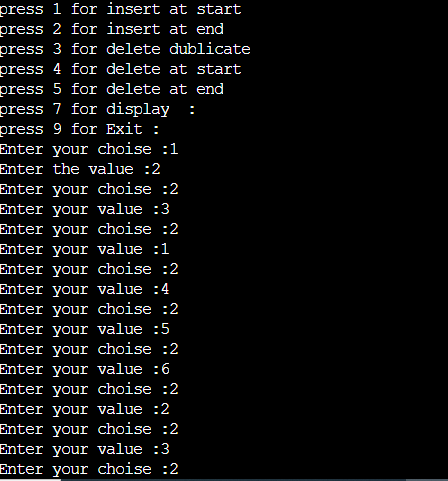
break;

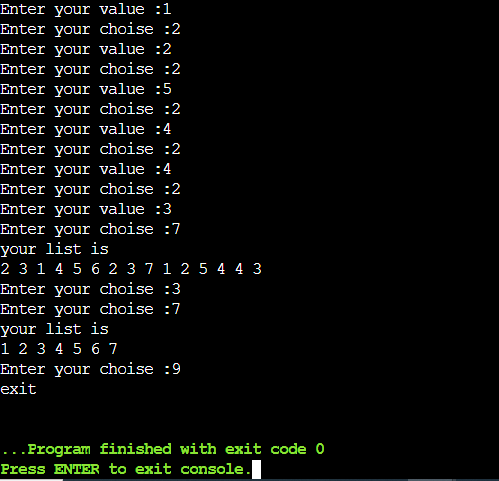
}

} while (ch != 9);

return 0;

}





**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| 1. | 05 |  |  |
| 2. | 05 |  |  |
| **Total** | **10** |  | **Signature** |

**Note : Attempt all tasks and get them checked by your Lab Instructor. Also for each task, attach a screenshot of the output.**