**Bahria University, Lahore Campus**

Department of Computer Science

Lab Journal 05

**(Spring 2023)**

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

Name: AFFAN AHMAD \_ Enroll No: 03-134221-003\_\_ Class: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 singly linked list data structure and perform following operations.

* Insert from start ()
* Insert from end ()
* Insert at any point ()
* Delete from start ()
* Delete from end ()
* Delete from any point ()
* 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;

struct node

{

int data;

node \*next;

node \*head, \*tail;

};

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 = temp;

}

}

void insertend(int value)

{

node \*temp = new node;

temp->data = value;

tail->next = temp;

tail = temp;

}

void insertposition(int pos, int value)

{ node \*pr;

node \*cur;

node \*temp = new node;

cur = head;

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

{

pr = cur;

cur = cur->next;

}

temp->data = value;

pr->next = temp;

temp->next = cur;

}

void deletefirst()

{

node \*temp;

temp = head;

head = head->next;

delete temp;

}

void deletelast()

{

node \*cur;

node \*pre;

cur=head;

while(cur->next!=NULL)

{

pre=cur;

cur=cur->next;

}

tail=pre;

pre->next=NULL;

delete cur;

}

void deleteposition(int pos)

{

node \*cur;

node \*pre;

cur=head;

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

{

pre=cur;

cur=cur->next;

}

pre->next=cur->next;

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 3 for insert in any position " << endl;

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

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

cout << "press 6 for delete at position " << endl;

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

cout << "press 8 for Exit :" << endl;

do{

cout << "Enter your choise :";

cin >> ch;

if (ch == 1)

{

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)

{

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

insertend(val);

}

if (ch == 4)

{

deletefirst();

}

if (ch == 5)

{

deletelast();

}

if(ch == 6)

{

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

deleteposition(pos);

}

if (ch == 3)

{

cout << "Enter your position :"; cin >> pos;

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

insertposition(pos,val);

}

if (ch == 8)

{

cout << "exit" << endl;

break;

}

} while (ch != 8);

return 0;

}

**Task 2**

Add additional function **count\_nodes ()** in Task 1. The function should count and display total nodes present in a linked list.

#include<iostream>

#include <stdlib.h>

using namespace std;

int t =0;

struct node

{

int data;

node \*next;

node \*head, \*tail;

};

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 = temp;

}

}

void insertend(int value)

{

node \*temp = new node;

temp->data = value;

tail->next = temp;

tail = temp;

}

void insertposition(int pos, int value)

{

node \*pr;

node \*cur;

node \*temp = new node;

cur = head;

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

{

pr = cur;

cur = cur->next;

}

temp->data = value;

pr->next = temp;

temp->next = cur;

}

void deletefirst()

{

node \*temp;

temp = head;

head = head->next;

delete temp;

}

void deletelast()

{

node \*cur;

node \*pre;

cur=head;

while(cur->next!=NULL)

{

pre=cur;

cur=cur->next;

}

tail=pre;

pre->next=NULL;

delete cur;

}

void deleteposition(int pos)

{

node \*cur;

node \*pre;

cur=head;

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

{

pre=cur;

cur=cur->next;

}

pre->next=cur->next;

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 3 for insert in any position " << endl;

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

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

cout << "press 6 for delete at position " << endl;

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

cout << "press 8 for total nodes :" << 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 == 6)

{

t=t-1;

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

deleteposition(pos);

}

if (ch == 3)

{

++t;

cout << "Enter your position :"; cin >> pos;

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

insertposition(pos,val);

}

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;

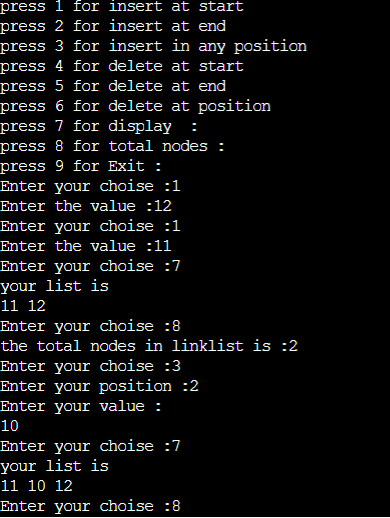
break;

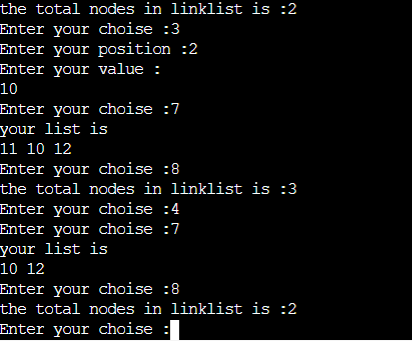
}

} while (ch != 9);

return 0;

}





**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| 1. | 07 |  |  |
| 2. | 03 |  |  |
| **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.**