**Experiment Number - 4**

**Student Name:** ANIKET KUMAR **UID:** 20BCS5306

**Branch:** CSE **Section/Group:** 20BCS\_WM-703 / B

**Semester:** 5th **Date of Performance:** 06th Sep, 2022

**Subject Name:** DAA LAB **Subject Code:** 20CSP-312

**1. Aim/Overview of the practical:** Code to Insert and Delete an element at the beginning and at the end in Doubly Linked List and Circular linked list.

**2. Software Requirements:**

Visual Studio Code (IDE).

**3. Program Code for Doubly Linked List:**

#include <iostream>

using namespace std;

class Node

{

public:

int data;

Node \*next;

Node \*prev;

};

void traversal(Node \*head)

{

while (head != NULL)

{

cout << head->data << " ";

head = head->next;

}

cout << "\n";

}

Node \*insertHead(Node \*head, int data)

{

Node \*newNode = new Node;

if (head == NULL)

return 0;

else

{

newNode->data = data;

head->prev = newNode;

newNode->next = head;

newNode->prev = NULL;

head = newNode;

return head;

}

}

Node \*insertEnd(Node \*head, int data)

{

Node \*i = head;

Node \*newNode = new Node;

newNode->data = data;

if (head == NULL)

return 0;

else

{

while (i->next != NULL)

{

i = i->next;

}

newNode->next = NULL;

newNode->prev = i;

i->next = newNode;

return head;

}

}

Node \*deleteFirst(Node \*head)

{

Node \*i = head->next;

head = i;

free(i->prev);

i->prev = NULL;

return head;

}

Node \*deleteEnd(Node \*head)

{

Node \*j = head->next;

Node \*i = head;

while (j->next != NULL)

{

j = j->next;

i = i->next;

}

free(i->next);

i->next = NULL;

return head;

}

int main(int argc, char const \*argv[])

{

Node \*head = new Node;

Node \*second = new Node;

Node \*third = new Node;

Node \*fourth = new Node;

head->prev = NULL;

head->data = 3;

head->next = second;

second->prev = head;

second->data = 5;

second->next = third;

third->prev = second;

third->data = 7;

third->next = fourth;

fourth->prev = third;

fourth->data = 9;

fourth->next = NULL;

cout << "After insertion" << endl;

head = insertHead(head, 1);

head = insertEnd(head, 11);

traversal(head);

cout << "After deletion" << endl;

head = deleteFirst(head);

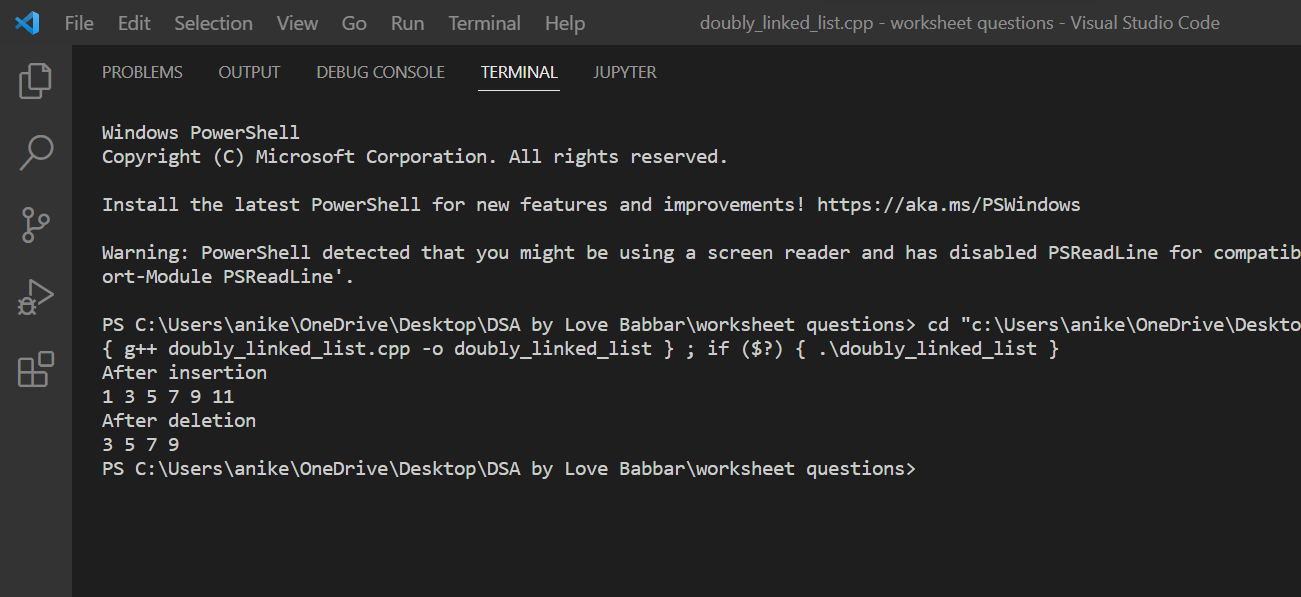
head = deleteEnd(head);

traversal(head);

return 0;

}

**4. OUTPUT:**

****

**5. Program Code for Circular Linked List:**

#include <iostream>

using namespace std;

struct Node

{

int data;

struct Node \*next;

};

struct Node \*insertInEmpty(struct Node \*last, int new\_data)

{

if (last != NULL)

return last;

struct Node \*temp = new Node;

temp->data = new\_data;

last = temp;

last->next = last;

return last;

}

struct Node \*insertAtBegin(struct Node \*last, int new\_data)

{

if (last == NULL)

return insertInEmpty(last, new\_data);

struct Node \*temp = new Node;

temp->data = new\_data;

temp->next = last->next;

last->next = temp;

return last;

}

struct Node \*insertAtEnd(struct Node \*last, int new\_data)

{

if (last == NULL)

return insertInEmpty(last, new\_data);

struct Node \*temp = new Node;

temp->data = new\_data;

temp->next = last->next;

last->next = temp;

last = temp;

return last;

}

struct Node \*insertAfter(struct Node \*last, int new\_data, int after\_item)

{

if (last == NULL)

return NULL;

struct Node \*temp, \*p;

p = last->next;

do

{

if (p->data == after\_item)

{

temp = new Node;

temp->data = new\_data;

temp->next = p->next;

p->next = temp;

if (p == last)

last = temp;

return last;

}

p = p->next;

} while (p != last->next);

cout << "The node with data " << after\_item << " is not present in the list." << endl;

return last;

}

void traverseList(struct Node \*last)

{

struct Node \*p;

if (last == NULL)

{

cout << "Circular linked List is empty." << endl;

return;

}

p = last->next;

do

{

cout << p->data << "->";

p = p->next;

} while (p != last->next);

if (p == last->next)

cout << p->data;

cout << "\n\n";

}

void deleteNode(Node \*\*head, int key)

{

if (\*head == NULL)

return;

if ((\*head)->data == key && (\*head)->next == \*head)

{

free(\*head);

\*head = NULL;

}

Node \*last = \*head, \*d;

if ((\*head)->data == key)

{

while (last->next != \*head)

last = last->next;

last->next = (\*head)->next;

free(\*head);

\*head = last->next;

}

while (last->next != \*head && last->next->data != key)

{

last = last->next;

}

if (last->next->data == key)

{

d = last->next;

last->next = d->next;

cout << key << " is delete" << endl;

free(d);

cout << endl;

cout << "Circular linked list after deleting " << key << endl;

traverseList(last);

}

else

cout << "The node with data " << key << " not found in the list" << endl;

}

int main()

{

struct Node \*last = NULL;

last = insertInEmpty(last, 30);

last = insertAtBegin(last, 20);

last = insertAtBegin(last, 10);

last = insertAtEnd(last, 40);

last = insertAtEnd(last, 60);

last = insertAfter(last, 50, 40);

cout << "circular linked list is : " << endl;

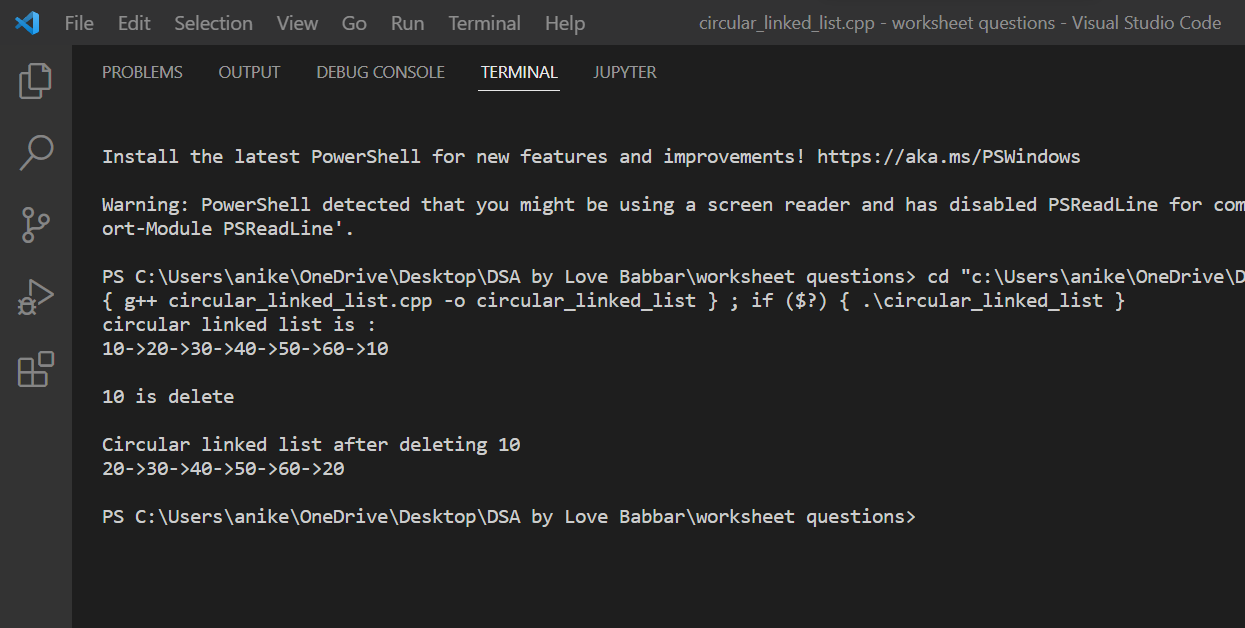
traverseList(last);

deleteNode(&last, 10);

return 0;

}

**6. OUTPUT:**

****

**Learning outcomes (What I have learnt):**

1. I have learnt how to write program in C++.
2. I have learnt about conditionals(if-else, switch-case) and loops(for loop, while loop, do-while loop).
3. I have learnt about arrays and strings.
4. I have learnt how to create functions in C++.
5. I have learnt about linked list, stacks, queues, etc.
6. I have also learnt about doubly linked list and circular linked list.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |