**Trisha Lakhani**

**1BM19CS214**

**DOUBLY LINKED LIST**

WAP Implement doubly link list with primitive operations a) Create a doubly linked list. b) Insert a new node to the left of the node. c) Delete the node based on a specific value d) Display the contents of the list

#include <stdio.h>

#include <stdlib.h>

struct node{

struct node \*prev;

int data;

struct node \*next;

};

struct node \*head=NULL;

void add\_at\_begin( ){

struct node \*ptr = NULL;

ptr=(struct node \*)malloc(sizeof(struct node));

printf("enter the node data :");

scanf("%d",& ptr->data);

ptr->prev=NULL;

ptr->next=NULL;

if(head==NULL){

head=ptr;

}

else{

ptr->next=head;

head->prev=ptr;

head=ptr;

}

}

void delete\_at\_specifiedloc( ) {

struct node \*ptr, \*temp;

int val;

printf("Enter the value after which the node will be deleted");

scanf("%d",&val);

temp = head;

while(temp -> data != val)

temp = temp -> next;

if(temp -> next == NULL) {

printf("\nCan't delete\n");

}

else if(temp -> next -> next == NULL)

{ temp ->next = NULL;

printf("\nNode Deleted\n");

} else

{ ptr = temp -> next;

temp -> next = ptr -> next;

ptr -> next -> prev = temp;

free(ptr);

printf("\nNode Deleted\n");

} }

void display( ){

if(head==NULL){

printf("list is empty\n");

}

else{

struct node \*temp=head;

while(temp!=NULL){

printf("%d\t",temp->data);

temp=temp->next;

}

printf("\n");

}

}

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

int opt=0;

while(1){

printf("DOUBLY LINKED LIST\n");

printf("1.add\_at\_begin\n");

printf("2.delete\_at\_pos\n");

printf("3.display\n");

printf("enter the option :");

scanf("%d",& opt);

switch(opt){

case 1:

add\_at\_begin();

break;

case 2:

delete\_at\_specifiedloc();

break;

case 3:

display();

break;

default:

printf("invalid option\n");

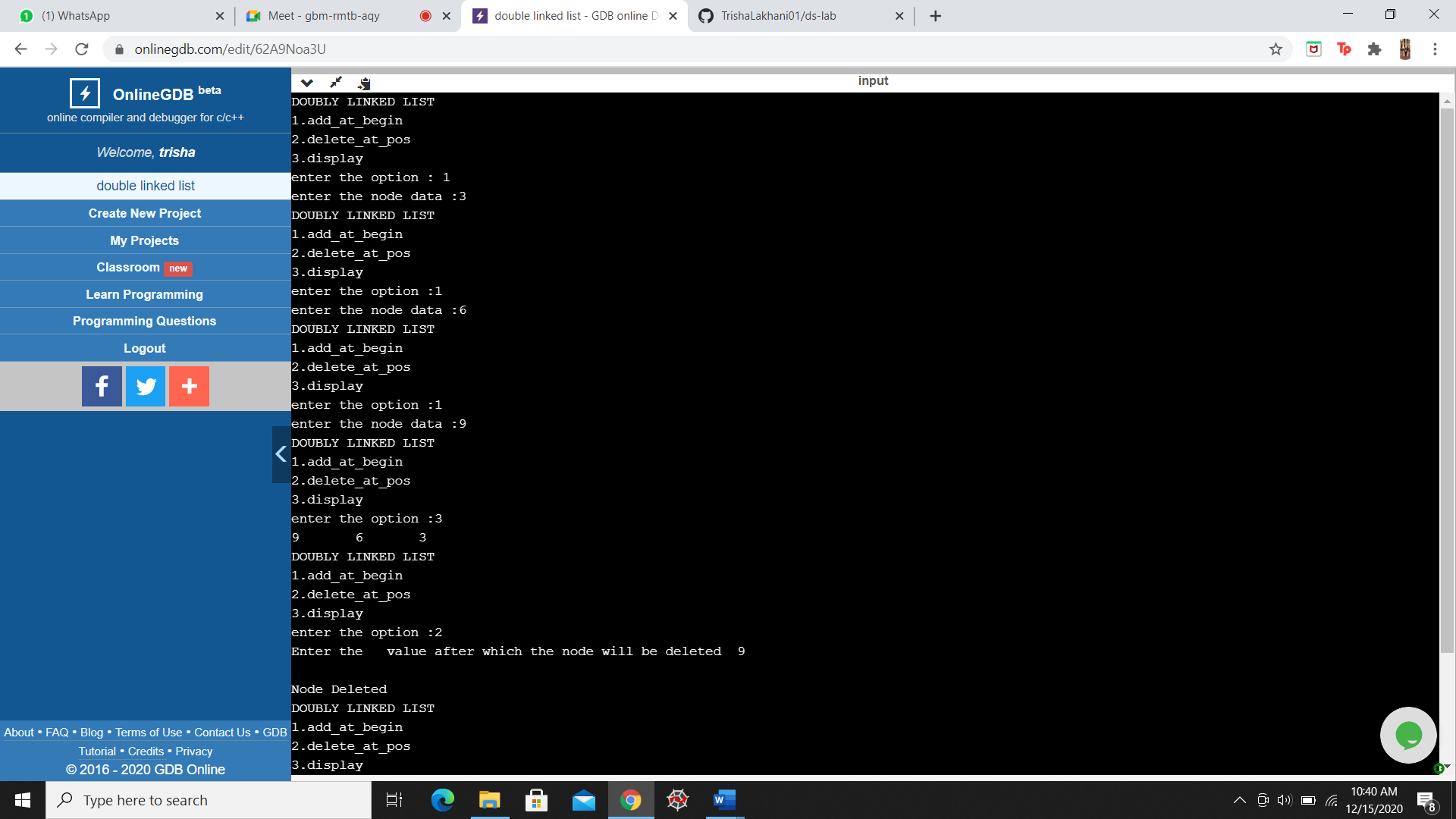
}

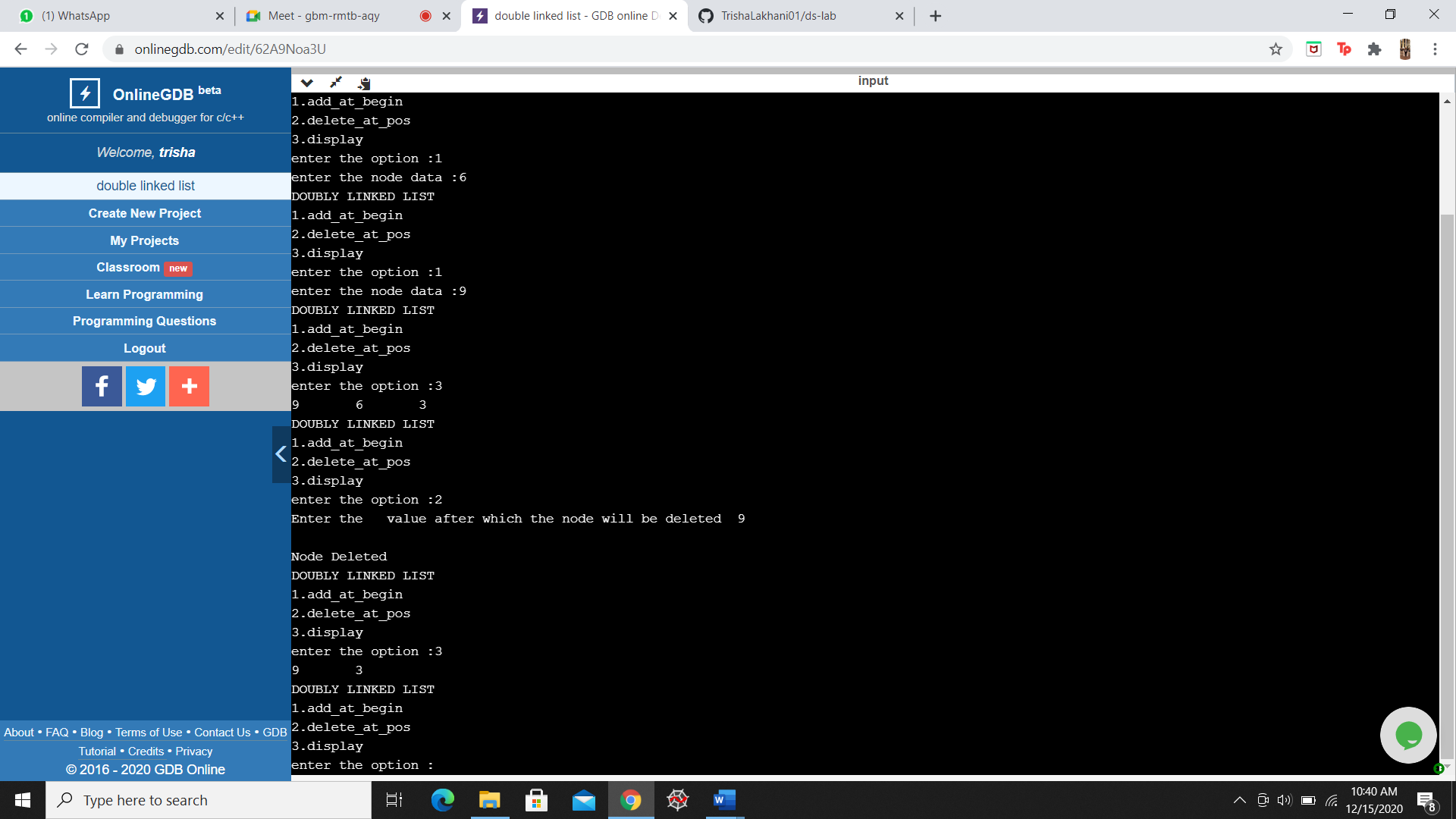
}

return 0;

}

**OUTPUT**

****

****