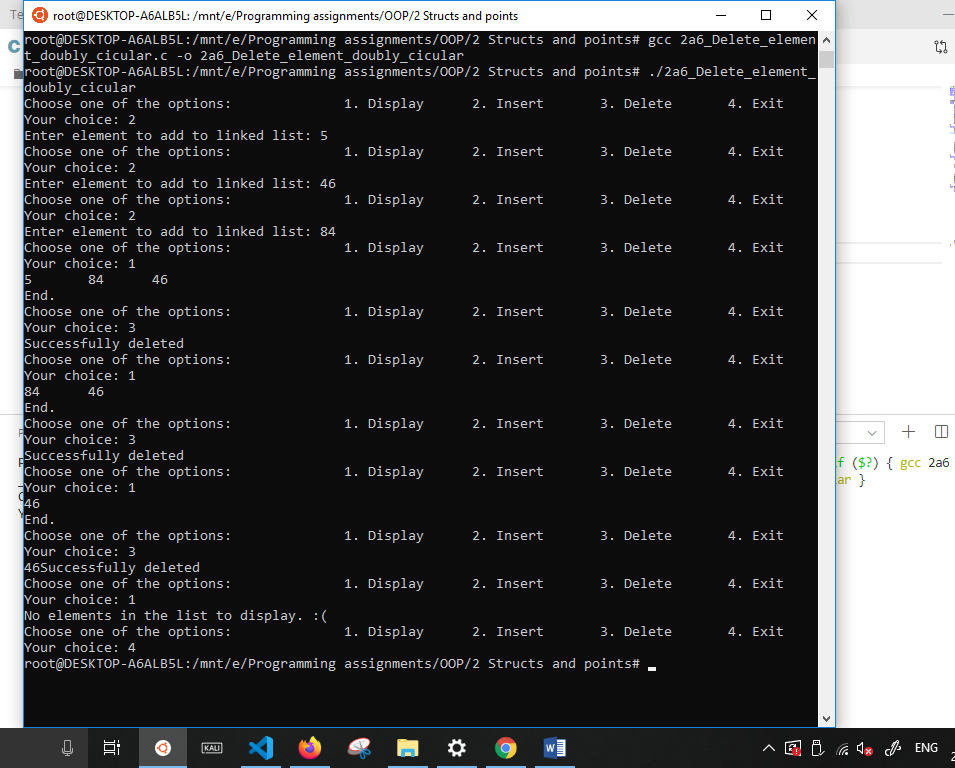
Output :



struct node\* delete(struct node \*handle){

    if(handle==NULL){

        printf("No elements to delete :(\n");

        return handle;

    }

    if(handle->next==handle){

        printf("%d",handle->next->data);

        handle = NULL;

        return handle;

    }

    handle->next->prev = handle->prev;

    handle->prev->next = handle->next;

    handle = handle->next;

    return handle;

}

int main(){

    int el, choice;

    while(1){

        printf("Choose one of the options:\t\t");

        printf("1. Display\t");

        printf("2. Insert\t");

        printf("3. Delete\t");

        printf("4. Exit\n");

        printf("Your choice: ");

        scanf("%d",&choice);

        switch (choice){

        case 1:

            display(handle);

            break;

        case 2:

            printf("Enter element to add to linked list: ");

            scanf("%d",&el);

            handle = insert(el,handle);

            break;

        case 3:

            handle = delete(handle);

            printf("Successfully deleted\n");

            break;

        default:

            return 0;

        }

    } return 0;

}

// Write a function to delete an element into Doubly Circular Linked List

#include<stdio.h>

#include<stdlib.h>

int el, pos;

struct node

{

    int data;

    struct node \*prev;

struct node \*next;

}\*handle, \*new, \*current;

void display(struct node \*handle){

    if(handle==NULL){

    printf("No elements in the list to display. :(\n");

    return;

    }

    current = handle;

    do{

    printf("%d\t",current->data);

    current = current->next;

    }while(current!=handle);

    printf("\nEnd.\n");

}

struct node\* insert(int ele,struct node \*handle){

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

    new ->data=ele;

    if(handle==NULL){

        printf("null");

        new ->next=new;

        new->prev=new;

        handle = new;

        return handle;

    }

    new->prev=handle;

    new->next=handle->next;

    handle->next->prev=new;

    handle->next=new;

    return handle;

}

Code :