Nithin S
221IT085

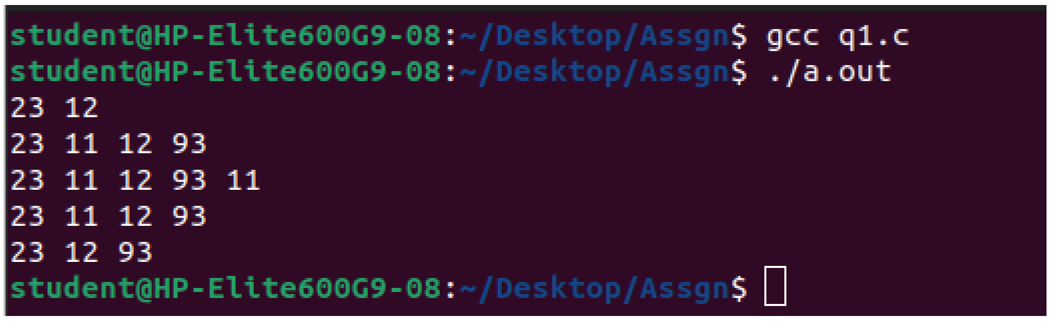
IT206 Lab Assignment 3

Q1) Implement a doubly linked list

```
#include<stdio.h>
#include<stdlib.h>

struct Node
{
    int data;
    struct Node* next;
    struct Node* prev;
};

int count(struct Node *head)
{

    int c=0;
    while(head)
    {
        c++;
        head=head->next;

    }
    return c;
}

void insert(struct Node **head,int x,int idx)
{
    if(idx<0 || idx>count(*head))
    {
        printf("Invalid Index.\n");
        return;
    }
    struct Node*temp=(struct Node *)malloc(sizeof(struct 
Node));
    temp->data=x;
    if(idx==0)
    {
        temp->prev=NULL;
        temp->next=*head;
        if(*head!=NULL) (*head)->prev=temp;
        *head=temp;
    }
    else
    {
        struct Node *p=*head;
        for(int i=0;i<idx-1;i++) p=p->next;
        temp->next=p->next;
        temp->prev=p;
        if(p->next) p->next->prev=temp;
        p->next=temp;
    }
}

int delete(struct Node**head,int idx)
{
    int x=-1;
    if(idx<0 || idx>count(*head)-1)
    {
        printf("Invalid Index\n");
        return x;
    }
    struct Node *p=*head;
    if(idx==0)
    {
        (*head)=(*head)->next;
        x=p->data;
        free(p);
        if(*head) (*head)->prev=NULL;
    }
    else
    {
        for(int i=0;i<idx;i++) p=p->next;
        x=p->data;
        p->prev->next=p->next;
        if(p->next) p->next->prev=p->prev;
        free(p);
    }
    return x;
}
```

```
void display(struct Node* head)
{
    while(head)
    {
        printf("%d ",head->data);
        head=head->next;
    }
    printf("\n");
}

int main()
{
    struct Node *head=NULL;
    //put index of the place to be deleted or inserted.
    insert(&head,23,0);
    insert(&head,12,1);
    display(head);
    insert(&head,93,2);
    insert(&head,11,1);
    display(head);
    insert(&head,11,4);
    display(head);
    delete(&head,4);
    display(head);
    delete(&head,1);
    display(head);
    return 0;
}
```

OUTPUT

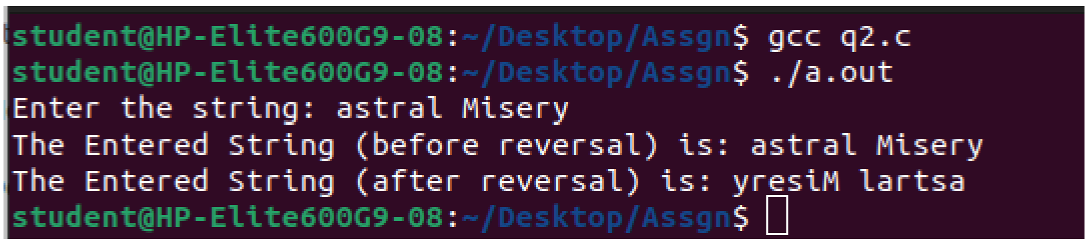


```
student@HP-Elite600G9-08:~/Desktop/Assgn$ gcc q1.c
student@HP-Elite600G9-08:~/Desktop/Assgn$ ./a.out
23 12
23 11 12 93
23 11 12 93 11
23 11 12 93
23 12 93
student@HP-Elite600G9-08:~/Desktop/Assgn$ 
```

Q2) Reverse a string using a doubly LinkedList

```
#include<stdio.h>
#include<stdlib.h>

struct Node
{
    char data;
    struct Node* next;
    struct Node* prev;
};

int count(struct Node *head)
{

    int c=0;
    while(head)
    {
        c++;
        head=head->next;

    }
    return c;
}

void insert(struct Node **head,char x,int idx)
{
    if(idx<0 || idx>count(*head))
    {
        printf("Invalid Index.\n");
        return;
    }
    struct Node*temp=(struct Node
*)malloc(sizeof(struct Node));
    temp->data=x;
    if(idx==0)
    {
        temp->prev=NULL;
        temp->next=*head;
        if(*head!=NULL) (*head)->prev=temp;
        *head=temp;
    }
    else
    {
        struct Node *p=*head;
        for(int i=0;i<idx-1;i++) p=p->next;
        temp->next=p->next;
        temp->prev=p;
        if(p->next) p->next->prev=temp;
        p->next=temp;
    }
}

void reverse(struct Node**head)
{
    struct Node *p=*head;
    struct Node *temp=NULL;
    while(p)
    {
        temp=p->prev;
        p->prev=p->next;
        p->next=temp;
        p=p->prev;
    }
    if(temp) (*head)=temp->prev;
}

void display(struct Node* head)
{
    while(head)
    {
        printf("%c",head->data);
        head=head->next;
    }
    printf("\n");
}

int main() {
    struct Node* head = NULL;
    printf("Enter the string: ");

    char ele;
    int count=0;

    while (1) {
        ele = getchar();
        if (ele == '\n') {
            break;
        }
        insert(&head, ele, count);
        count++;
    }

    printf("The Entered String (before reversal) is: 
");
    display(head);

    printf("The Entered String (after reversal) is: 
");
    reverse(&head);
    display(head);

    return 0;
}
```

OUTPUT

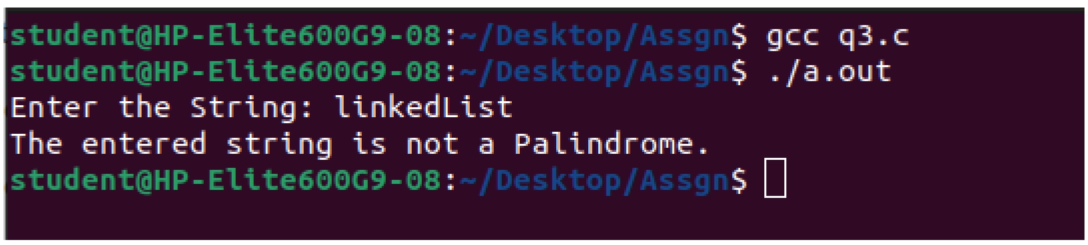


```
student@HP-Elite600G9-08:~/Desktop/Assgn$ gcc q2.c
student@HP-Elite600G9-08:~/Desktop/Assgn$ ./a.out
Enter the string: astral Misery
The Entered String (before reversal) is: astral Misery
The Entered String (after reversal) is: yresiM lartsa
student@HP-Elite600G9-08:~/Desktop/Assgn$ 
```

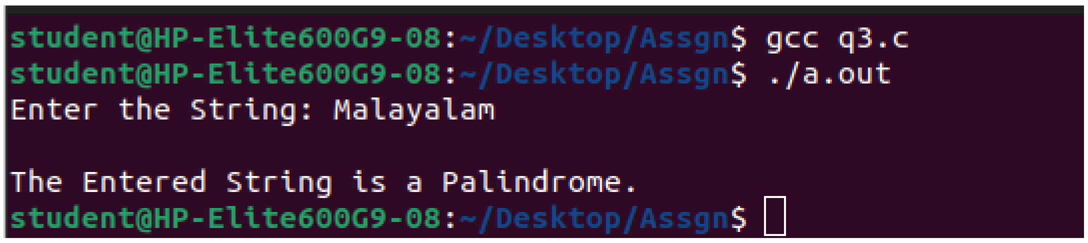
Q3) Check whether a string is palindrome using a doubly Linked List

```
#include<stdio.h>
#include<stdlib.h>
#include <stdbool.h>
#include<ctype.h>

struct Node
{
    char data;
    struct Node* next;
    struct Node* prev;
};

int count(struct Node *head)
{

    int c=0;
    while(head)
    {
        c++;
        head=head->next;
    }
    return c;
}

void insert(struct Node **head,char x,int idx)
{
    if(idx<0 || idx>count(*head))
    {
        printf("Invalid Index.\n");
        return;
    }
    struct Node*temp=(struct Node
*)malloc(sizeof(struct Node));
    temp->data=x;
    if(idx==0)
    {
        temp->prev=NULL;
        temp->next=*head;
        if(*head!=NULL) (*head)->prev=temp;
        *head=temp;
    }
    else
    {
        struct Node *p=*head;
        for(int i=0;i<idx-1;i++) p=p->next;
        temp->next=p->next;
        temp->prev=p;
        if(p->next) p->next->prev=temp;
        p->next=temp;
    }
}

void display(struct Node* head)
{
    while(head)
    {
        printf("%c",head->data);
        head=head->next;
    }
    printf("\n");
}

bool isPalindrome(struct Node  *head)
{
    struct Node *start=head;
    struct Node *end=head;
    if(start==NULL || start->next==NULL) return true;
    while(end->next)
    {
        end=end->next;
    }
    while(start!=end)
    {
        if(start->data != end->data) return false;
        if(start->next==end) break;
        start=start->next;
        end=end->prev;
    }
    return true;
}


int main() {
    struct Node* head = NULL;
    printf("Enter the String: ");

    char ele;
    int count = 0;

    while (1) {
        ele = tolower(getchar());
        if (ele == '\n') {
            break;
        }
        insert(&head, ele, count);
        count++;
    }

    if (isPalindrome(head)) {
        printf("\nThe Entered String is a
Palindrome.\n");
    } else {
        printf("The entered string is not a
Palindrome.\n");
    }

    return 0;
}
```

OUTPUT



```
student@HP-Elite600G9-08:~/Desktop/Assgn$ gcc q3.c
student@HP-Elite600G9-08:~/Desktop/Assgn$ ./a.out
Enter the String: linkedList
The entered string is not a Palindrome.
student@HP-Elite600G9-08:~/Desktop/Assgn$ 
```



```
student@HP-Elite600G9-08:~/Desktop/Assgn$ gcc q3.c
student@HP-Elite600G9-08:~/Desktop/Assgn$ ./a.out
Enter the String: Malayalam

The Entered String is a Palindrome.
student@HP-Elite600G9-08:~/Desktop/Assgn$ 
```