**NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR**

**Cachar, Assam**

**B.Tech. IIIrd Sem**

**Subject Code:** CS211

**Submitted By:**

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

/\*

Q. 1. Lab Question: Write a program to insert a node into a singly linked list at front, before a node, after a node, and last.

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct NODE

{

    int value;

    struct NODE \*next;

}\*front = NULL, \*rear = NULL;

void PUSH ()

{

    struct NODE \*temp = (struct NODE\*) malloc (sizeof (struct NODE));

    int num;

    printf ("Enter Value to be added: ");

    scanf ("%d", &num);

    temp->value = num;

    if (front == NULL)

    {

        temp->next = NULL;

        front = temp;

    }

    else

    {

        rear -> next = temp;

    }

    rear = temp;

    printf ("New node added...\n");

}

void TRAVERSE ()

{

    struct NODE \*temp = front;

    if (front == NULL)

        printf ("Queue Empty...\n");

    else

    {

        while (temp != rear)

        {

            printf (" %d ",temp -> value);

            temp = temp->next;

        }

        printf ("%d\n", temp -> value);

    }

}

int main()

{

    char yn['!'];

    printf ("Firstly Input Nodes.\n");

    do

    {

        PUSH();

        printf ("\nEnter more nodes? y/n...\n>");

        scanf ("%s", &yn);

    } while (strcmpi (yn, "n"));

    printf ("\nYour created Nodes: ");

    TRAVERSE ();

    int option;

    do

    {

        printf ("\nSelect the option: ");

        printf ("\n1. Add New Node at Front");

        printf ("\n2. Add New Node Before an existing Node");

        printf ("\n3. Add New Node After an existing Node");

        printf ("\n4. Add Node at Last");

        printf ("\n5. Exit\n\n> ");

        scanf ("%d", &option);

        printf ("\n");

        if (option==5)

            break;

        struct NODE \*newnode = (struct NODE\*) malloc (sizeof (struct NODE));

        int num, existing;

        struct NODE \*temp = front;

        printf ("Enter Value to be Added: ");

            scanf ("%d", &num);

        newnode->value = num;

        switch (option)

        {

            case 1 :

                    newnode->next = front;

                    front = newnode;

                    break;

            case 2 :

                    printf ("Choose data of existing node: ");

                    scanf ("%d", &existing);

                    while (temp != rear)

                    {

                        if ((temp->next)->value == existing);

                        {

                            newnode->next = (temp->next)->next;

                            temp->next = newnode;

                            break;

                        }

                        temp=temp->next;

                    }

                    break;

            case 3 :

                    printf ("Choose data of existing node: ");

                    scanf ("%d", &existing);

                    while (temp != rear)

                    {

                        if ((temp->next)->value == existing);

                        {

                            newnode->next = ((temp->next)->next)->next;

                            (temp->next)->next = newnode;

                            break;

                        }

                        temp=temp->next;

                    }

                    break;

            case 4 :

                    newnode->next = NULL;

                    rear -> next = newnode;

                    rear = newnode;

                    break;

            default :

                    break;

        }

    }  while (option < 5 && option > 0);

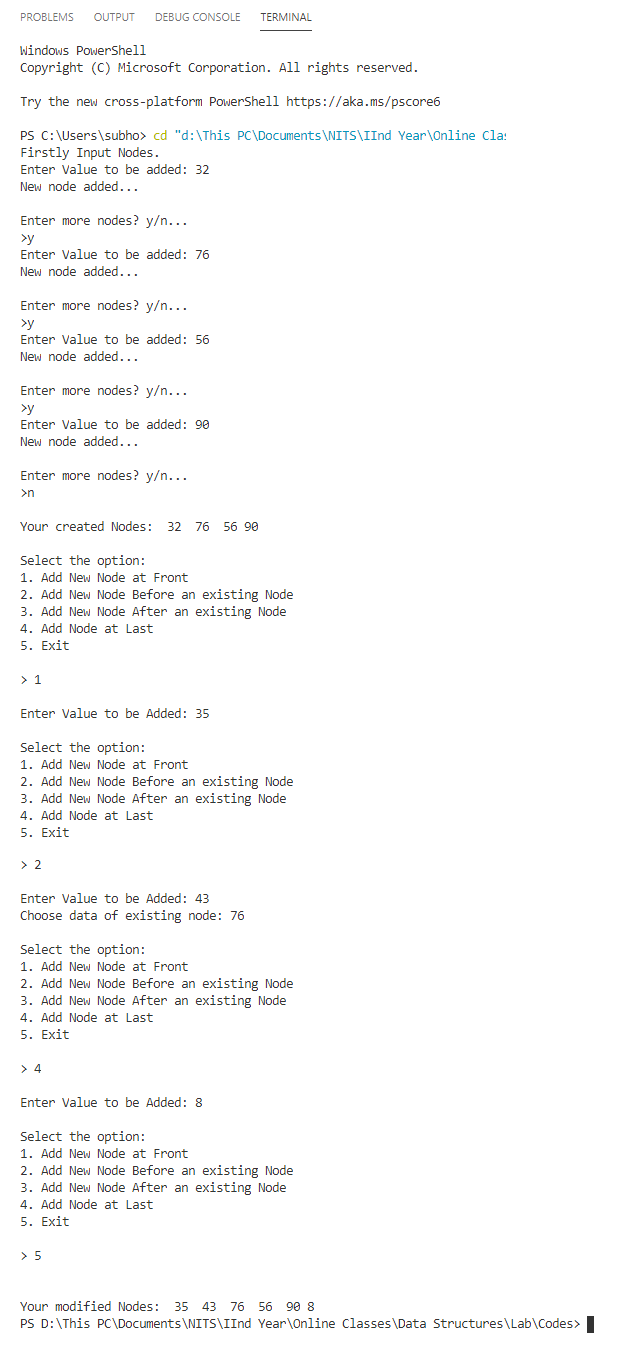
    printf ("\nYour modified Nodes: ");

    TRAVERSE ();

    return 0;

}

OUTPUT:

/\*

Q. 2. Lab Question: Write a program to delete an item from a singly linked list.

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct NODE

{

    int value, position;

    struct NODE \*next;

}\*front = NULL, \*rear = NULL;

int PUSH (int pos)

{

    struct NODE \*temp = (struct NODE\*) malloc (sizeof (struct NODE));

    temp->position = pos;

    int num;

    printf ("Enter Value to be added: ");

    scanf ("%d", &num);

    temp->value = num;

    if (front == NULL)

    {

        temp->next = NULL;

        front = temp;

    }

    else

    {

        rear -> next = temp;

    }

    rear = temp;

    printf ("New node added...\n");

    return 0;

}

int POP (int NodePosition)

{

    if (NodePosition == front->position)

    {

        front=front->next;

        printf ("Requested Node Deleted");

    }

    else

    {

        struct NODE \*temp = front;

        while (temp != rear)

        {

            if ((temp->next)->position == NodePosition)

            {

                temp->next = (temp->next)->next;

                printf ("Requested Node Deleted");

            }

            temp = temp->next;

        }

    }

    return 0;

}

void TRAVERSE ()

{

    struct NODE \*temp = front;

    if (front == NULL)

        printf ("Queue Empty...\n");

    else

    {

        while (temp != rear)

        {

            printf (" %d ",temp -> value);

            temp = temp->next;

        }

        printf ("%d\n", temp -> value);

    }

}

int main()

{

    int DeleteNodePosition, size=0, pos=0;

    char yn['!'];

    printf ("Firstly Input Nodes.\n");

    do

    {

        ++pos;

        PUSH(pos);

        printf ("\nEnter more nodes? y/n...\n>");

        scanf ("%s", &yn);

    } while (strcmpi (yn, "n"));

    printf ("Your created list is: ");

    TRAVERSE();

    printf ("Which Node do you wish to delete? Write its position: ");

    scanf ("%d", &DeleteNodePosition);

    POP (DeleteNodePosition);

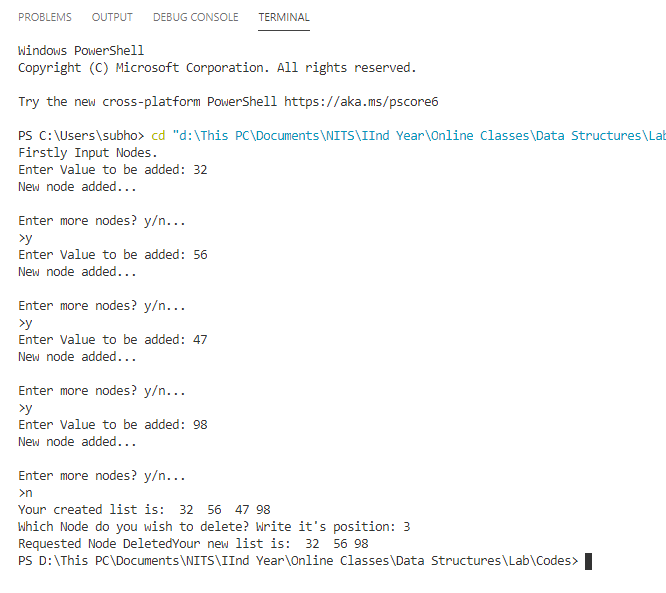
    printf ("Your new list is: ");

    TRAVERSE();

    return 0;

}

OUTPUT:



/\*

Q.3. Lab Question: Write a program to add two polynomial equations using single linked list.

\*/

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int coef,expo;

    struct node\* next;

};

struct node\* insertpoly(struct node\* thead,int c,int e)

{

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

    newnode->coef=c;

    newnode->expo=e;

    if(thead==NULL)

    {

        newnode->next=NULL;

        return newnode;

    }

    struct node\* prev,\* curr;

    prev=curr=thead;

    while(curr!=NULL && curr->expo>e){

        prev=curr;

        curr=curr->next;

    }

    if(curr==thead)

    {

        newnode->next=curr;

        return newnode;

    }

    else if(curr==NULL)

    {

        prev->next=newnode;

        newnode->next=NULL;

    }

    else{

        newnode->next=curr;

        prev->next=newnode;

    }

    return thead;

}

struct node\* append(struct node\* thead,int c,int e)

{

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

    newnode->coef=c;

    newnode->expo=e;

    if(thead==NULL)

    {

        newnode->next=NULL;

        return newnode;

    }

    struct node\* trav=thead;

    while(trav->next!=NULL)

        trav=trav->next;

    trav->next=newnode;

    newnode->next=NULL;

    return thead;

}

struct node\* polyaddition(struct node\* p1thead,struct node\* p2thead)

{

    struct node\* ans=NULL;

    struct node\* t1,\* t2;

    t1=p1thead;

    t2=p2thead;

    while(t1!=NULL && t2!=NULL){

        if(t1->expo > t2->expo){

            ans=append(ans,t1->coef,t1->expo);

            t1=t1->next;

        }

        else if(t1->expo < t2->expo){

            ans=append(ans,t2->coef,t2->expo);

            t2=t2->next;

        }

        else{

            ans=append(ans,(t1->coef)+(t2->coef),t1->expo);

            t1=t1->next;

            t2=t2->next;

        }

    }

    while(t1!=NULL)

    {

        ans=append(ans,t1->coef,t1->expo);

        t1=t1->next;

    }

    while(t2!=NULL)

    {

        ans=append(ans,t2->coef,t2->expo);

        t2=t2->next;

    }

    return ans;

}

void display(struct node\* thead)

{

    printf ("\nThe final added polynomial is: ");

    struct node\* temp=thead;

    if(temp==NULL){

        printf("\nEmpty..");

    }

    else{

        while(temp->next!=NULL){

            if(temp->coef == 0);

            else if(temp->coef == 1 && temp->expo == 1)

                printf("X+");

            else if(temp->coef == 1 && temp->expo != 0 && temp->expo != 1)

                printf("X^%d+",temp->expo);

            else if( temp->expo == 1)

                printf("%dX+",temp->coef);

            else

            printf("%dX^%d+",temp->coef,temp->expo);

            temp=temp->next;

        }

        if(temp->coef == 0);

        else if(temp->coef == 1 && temp->expo == 1)

            printf("X");

        else if(temp->coef == 1 && temp->expo != 0 && temp->expo != 1)

            printf("X^%d",temp->expo);

        else if( temp->expo == 0)

            printf("%d",temp->coef);

        else if( temp->expo == 1)

            printf("%dX",temp->coef);

        else

        printf("%dX^%d",temp->coef,temp->expo);

    }

}

int main() {

    int a,b,n,i;

    struct node\* p1head,\* p2head,\* p3head;

    p1head=p2head=NULL;

    printf ("Enter Number of Terms in First Polynomial: ");

    scanf("%d",&n);

    for(i=0;i<n;i++)

    {

        printf ("Coefficient ");

        scanf("%d",&a);

        printf ("Exponent");

        scanf("%d",&b);

        p1head=insertpoly(p1head,a,b);

    }

    printf ("Enter Number of Terms in Second Polynomial: ");

    scanf("%d",&n);

    for(i=0;i<n;i++)

    {

        printf ("Coefficient ");

        scanf("%d",&a);

        printf ("Exponent ");

        scanf("%d",&b);

        p2head=insertpoly(p2head,a,b);

    }

    p3head=polyaddition(p1head,p2head);

    display(p3head);

    return 0;

}

OUTPUT:

