**NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR**

**Cachar, Assam**

**B.Tech. IIIrd Sem**

**Subject Code:** CS211

**Date:** 24/11/2020

**Submitted By:**

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

/\*

**Q. 1. Lab Question: Write a program to insert a node into doubly 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, \*prev;

}\*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;

    temp -> next = NULL;

    if (front == NULL)

    {

        temp -> prev = NULL;

        front = temp;

    }

    else

    {

        rear -> next = temp;

        temp -> prev = rear;

    }

    rear = temp;

    printf ("New node added...\n");

}

int insert (int option, int position)

{

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

    int num, i;

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

        scanf ("%d", &num);

    newnode -> value = num;

    struct NODE \*point = front;

    for (i=1; i<position; ++i)

    {

        point = point -> next;

    }

    if (option == 1)

    {

        newnode->next = front;

        front -> prev = newnode;

        front = newnode;

    }

    else if (option == 2)

    {

        (point -> prev) -> next = newnode;

        newnode -> prev = point -> prev;

        point -> prev = newnode;

        newnode -> next = point;

    }

    else if (option == 3)

    {

        newnode -> next = point -> next;

        newnode -> prev = point;

        (point -> next) -> prev = newnode;

        point -> next = newnode;

    }

    else if (option == 4)

    {

        newnode -> next = NULL;

        newnode -> prev = rear;

        rear -> next = newnode;

        rear = newnode;

    }

    else

    {

        printf ("Exiting...");

    }

    return 0;

}

int 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);

    }

    return 0;

}

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"));

int option;

do

{

    printf ("\nYour created Nodes: ");

    TRAVERSE ();

    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");

    int position = 1;

    if (option == 2 || option == 3)

    {

        printf ("Enter position for the existing in-respect node: ");

        scanf ("%d", &position);

    }

    insert (option, position);

} while (option < 5 && option > 0);

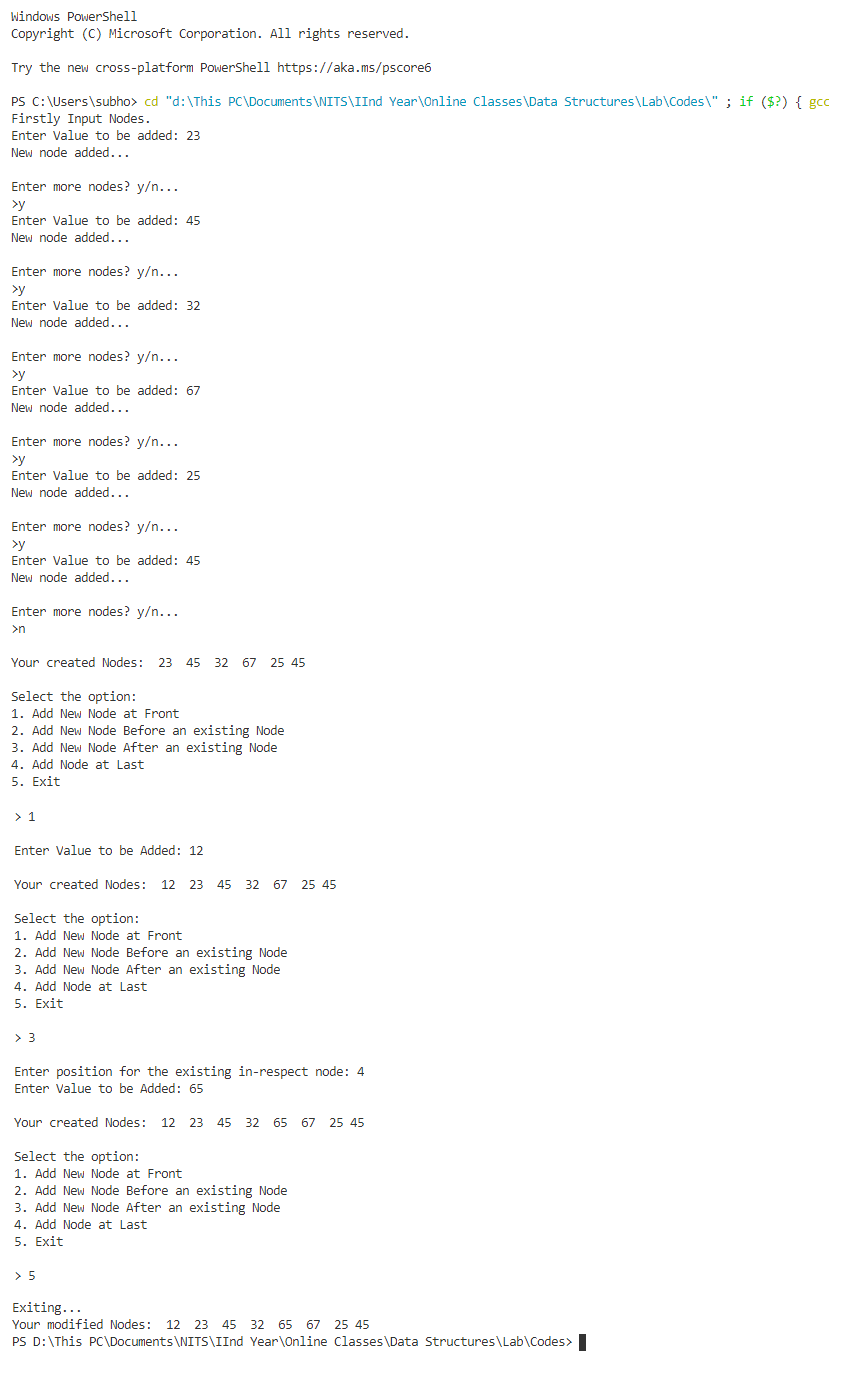
    printf ("\nYour modified Nodes: ");

    TRAVERSE ();

    return 0;

}

**OUTPUT:**



/\*

**Q. 2. Lab Question: Write a program to delete an item from a doubly linked list based on value provided by the user.**

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct NODE

{

    int value;

    struct NODE \*next, \*prev;

}\*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;

    temp -> next = NULL;

    if (front == NULL)

    {

        temp -> prev = NULL;

        front = temp;

    }

    else

    {

        rear -> next = temp;

        temp -> prev = rear;

    }

    rear = temp;

    printf ("New node added...\n");

}

int POP (int user\_position, int size)

{

    if (!user\_position)

    {

        front = rear = NULL;

    }

    else

    {

        if(user\_position < ( (size + 1) / 2))

        {

            struct NODE \*point = front;

            for (int i=0; i < user\_position; ++i, point = point -> next);

            if (point == front)

            {

                front = rear = NULL;

            }

            else

            {

                (point -> next) -> prev = point -> prev;

                (point -> prev) -> next = point -> next;

            }

        }

        else

        {

            struct NODE \*point = rear;

            for (int i = size; i > user\_position; --i, point = point -> prev);

            if (point == rear)

            {

                (rear -> prev) -> next = NULL;

                rear = rear -> prev;

            }

            else

            {

                (point -> next) -> prev = point -> prev;

                (point -> prev) -> next = point -> 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;

    char yn['!'];

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

    do

    {

        ++size;

        PUSH();

        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 it's position (Enter 0 if you wish to delete the entire node) :  ");

    scanf ("%d", &DeleteNodePosition);

    POP (DeleteNodePosition, size);

    printf ("Your new list is: ");

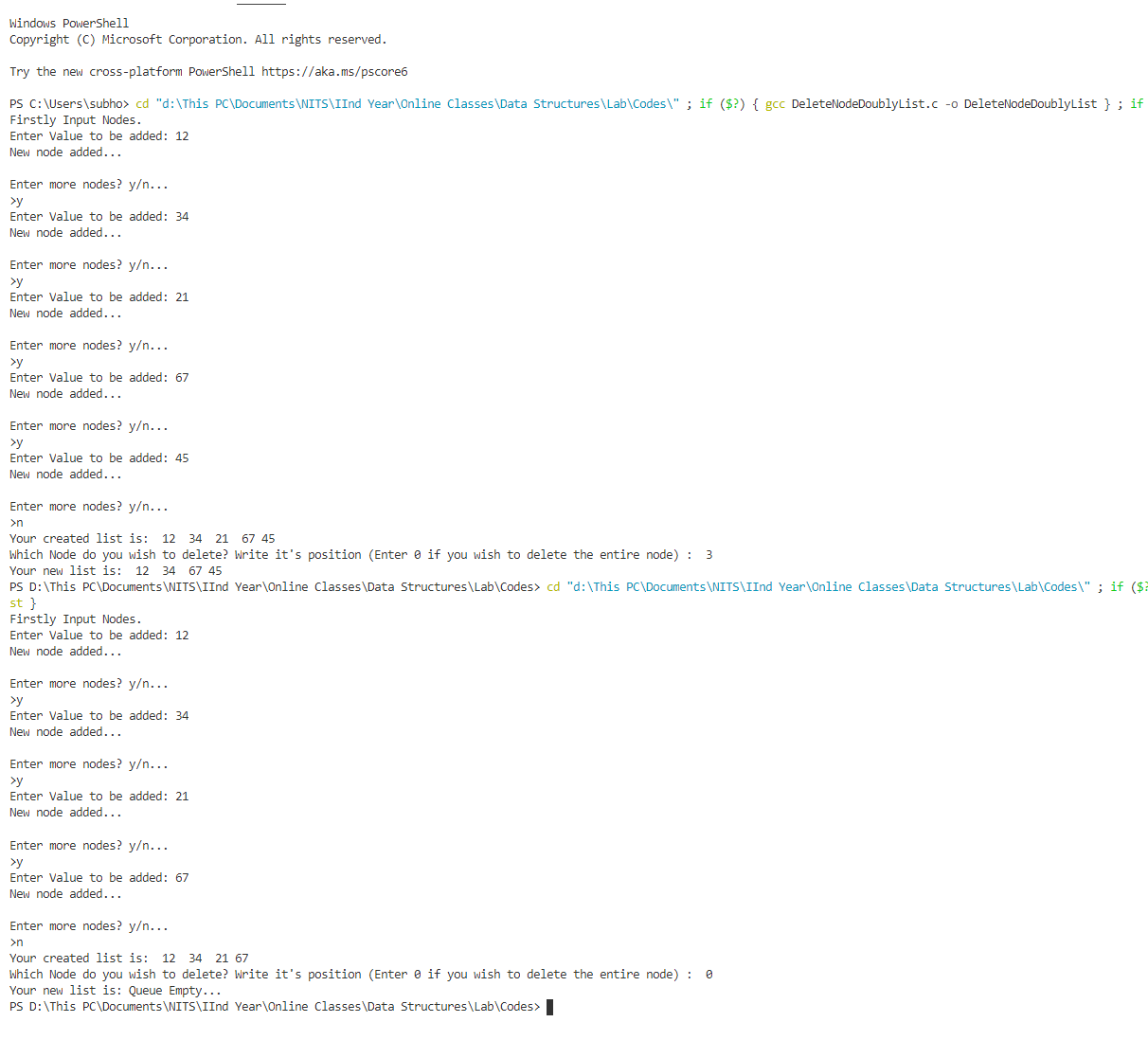
    TRAVERSE();

    return 0;

}

/\*

**OUTPUT:**



**\*/**

**Q. 3. Lab Question: Write a program to insert a node into circular linked list and also delete a node based on value provided by the user.**

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct NODE

{

    int value;

    struct NODE \*next;

}\*front = NULL, \*rear = NULL;

int 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)

    {

        front = temp;

    }

    else

    {

        rear -> next = temp;

    }

    temp -> next = front;

    rear = temp;

    printf ("New node added...\n");

    return 0;

}

int POP (int NodePosition)

{

    struct NODE \*temp = front;

    for (int i=1; i< (NodePosition-1); ++i)

    {

        temp = temp->next;

    }

    if (NodePosition == 1)

    {

        front=front->next;

        rear -> next = front;

        printf ("Requested Node Deleted\n");

    }

    else if (temp -> next == rear)

    {

        temp -> next = rear -> next;

        rear = temp;

        printf ("Requested Node Deleted\n");

    }

    else

    {

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

        printf ("Requested Node Deleted\n");

    }

    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", temp -> value);

    }

    temp = temp -> next;

    printf (" -> %d\n", temp->value);

}

int main()

{

    int DeleteNodePosition;

    char yn['!'];

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

    do

    {

        PUSH();

        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 it's position: ");

    scanf ("%d", &DeleteNodePosition);

    POP (DeleteNodePosition);

    printf ("Your new list is: ");

    TRAVERSE();

    return 0;

}

**OUTPUT:**

