**NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR**

**Cachar, Assam**

**B.Tech. IIIrd Sem**

**Subject Code:** CS211

**Date:** 27/11/2020

**Submitted By:**

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

/\*

**Q. 1. Lab Question: Write a program in C to insert and delete a middle element of a stack.**

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct NODE

{

    int value;

    struct NODE \*next;

}\*top = NULL, \*bottom = NULL;

int PUSH (int position)

{

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

    int num;

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

    scanf ("%d", &num);

    newnode->value = num;

    if (!position)

    {

        if (top == NULL)

        {

            newnode->next = NULL;

            bottom = newnode;

        }

        else

        {

            newnode -> next = top;

        }

        top = newnode;

    }

    else

    {

        struct NODE \*point = top;

        for (int i=0; i < position-1; ++i)

        {

            point = point -> next;

        }

        newnode -> next = point -> next;

        point -> next = newnode;

    }

    printf ("New node stacked...\n");

    return 0;

}

int POP (int position)

{

    struct NODE \*point = top;

    for (int i=0; i < (position - 1); ++i)

    {

        point = point -> next;

    }

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

    return 0;

}

int TRAVERSE ()

{

    struct NODE \*temp = top;

    if (top == NULL)

        printf ("Stack Empty...\n");

    else

    {

        printf ("Values are: ");

        while (temp != bottom)

        {

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

            temp = temp->next;

        }

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

    }

    return 0;

}

int main()

{

    int NodePosition, size=0, pos=0;

    char yn['!'];

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

    do

    {

        ++pos;

        PUSH(0);

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

        scanf ("%s", &yn);

    } while (strcmpi (yn, "n"));

    printf ("Your created list is: ");

    TRAVERSE();

    printf ("Adding new node at the middle location ");

    PUSH (pos/2);

    ++pos;

    TRAVERSE();

    printf ("Deleting Node from the middle position\n");

    POP (pos/2);

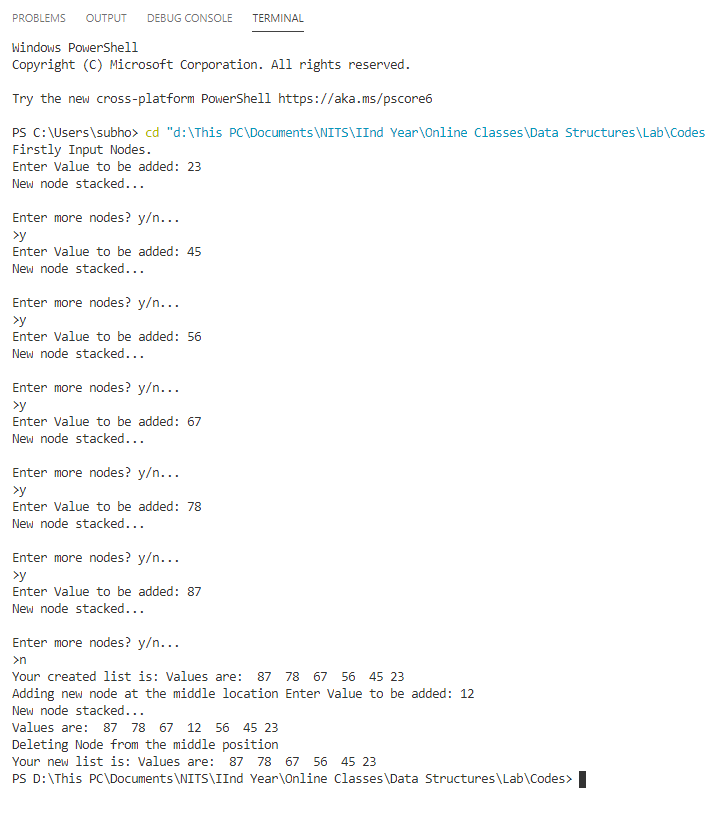
    printf ("Your new list is: ");

    TRAVERSE();

    return 0;

}

**OUTPUT:**



/\*

**Q. 1. Lab Question: Write a program in C to implement two stacks in an array. .**

\*/

#include <stdio.h>

#include <stdlib.h>

struct NODE

{

    int value;

    struct NODE \*next;

}\*top [2], \*bottom [2];

void PUSH (int stack\_no)

{

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

    int num;

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

    scanf ("%d", &num);

    temp->value = num;

    if (top [stack\_no] == NULL)

    {

        temp->next = NULL;

        bottom [stack\_no] = temp;

    }

    else

    {

        temp -> next = top [stack\_no];

    }

    top [stack\_no]= temp;

    printf ("New node added...\n");

}

void POP (int stack\_no)

{

    if (top [stack\_no] == NULL)

    {

        printf ("\nSTACK UNDERFLOW\n");

    }

    else if (top [stack\_no] -> next == NULL)

    {

        top [stack\_no] -> next = NULL;

        top [stack\_no] = bottom [stack\_no] = NULL;

        printf ("Existing top node deleted...\n");

    }

    else

    {

        top [stack\_no] = top [stack\_no] -> next;

        printf ("Existing top node deleted...\n");

    }

}

void TRAVERSE (int stack\_no)

{

    struct NODE \*temp = top [stack\_no];

    if (top [stack\_no]== NULL)

        printf ("Stack Empty...\n");

    else

    {

        printf ("Values are: ");

        while (temp != bottom [stack\_no])

        {

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

            temp = temp->next;

        }

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

    }

}

int main()

{

    top [0] = top [1] = bottom [0] = bottom [1] = NULL;

    int option;

    while (option != 7)

    {

        printf ("\nSelect the option: ");

        printf ("\n1. PUSH in Stack I");

        printf ("\n2. POP in Stack I");

        printf ("\n3. TRAVERSE in Stack I");

        printf ("\n4. PUSH in Stack II");

        printf ("\n5. POP in Stack II");

        printf ("\n6. TRAVERSE in Stack II");

        printf ("\n7. Exit\n\n> ");

        scanf ("%d", &option);

        printf ("\n");

        if(option == 1)

            PUSH (0);

        else if (option == 2)

            POP (0);

        else if (option == 3)

            TRAVERSE (0);

        else if(option == 4)

            PUSH (1);

        else if (option == 5)

            POP (1);

        else if (option == 6)

            TRAVERSE (1);

        else if (option == 7)

            break;

        else

            printf ("Invalid Selection... Try Again...\n");

    }

    return 0;

}

**OUTPUT:**

