**NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR**

**Cachar, Assam**

**B.Tech. IIIrd Sem**

**Subject Code:** CS211

**Date:** 04/12/2020

**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 queue.**

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct QUEUE

{

    int data;

    struct QUEUE \*next;

} \*begin = NULL, \*end = NULL;

int PUSH (int flag)

{

    struct QUEUE \*NewQueue = (struct QUEUE\*) malloc (sizeof (struct QUEUE));

    printf ("\nEnter Value: ");

    scanf ("%d", &(NewQueue -> data));

    if (flag == 0)

    {

        if (begin == NULL)

        {

            begin = NewQueue;

        }

        else

        {

            end -> next = NewQueue;

        }

        end = NewQueue;

        end -> next = NULL;

    }

    else

    {

        struct QUEUE \*point = begin;

        for (int i=1; i<flag; ++i)

        {

            point = point -> next;

        }

        NewQueue -> next = point -> next;

        point -> next = NewQueue;

    }

    return 0;

}

int POP (int flag)

{

    if (begin == NULL)

    {

        printf ("\nQUEUE UNDERFLOW\n");

    }

    else

    {

        struct QUEUE \*point = begin;

        for (int i=1; i<flag; ++i)

        {

            point = point -> next;

        }

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

    }

    return 0;

}

int TRAVERSE ()

{

    if (begin == NULL)

    {

        printf ("\nQueue Empty\n");

    }

    else

    {

        printf ("\nQueue: ");

        struct QUEUE \*point = begin;

        while (point != end)

        {

            printf ("%d ", point -> data);

            point = point -> next;

        }

        printf ("%d\n", point -> data);

    }

    return 0;

}

int main()

{

    char ch['!'];

    int count = 0;

    printf ("Firstly, Enter values into the Queue\n");

    do

    {

        ++count;

        PUSH (0);

        printf ("\nEnter more? Y/N...\n> ");

        scanf ("%s", &ch);

    } while (strcmpi (ch, "n"));

    TRAVERSE ();

    printf ("\nQueue Successfully Created...\nFinding middle of the Queue to add value...\n");

    PUSH (count/2);

    TRAVERSE ();

    printf ("\nFinding middle of the Queue to delete the value...\n");

    ++count;

    POP (count/2);

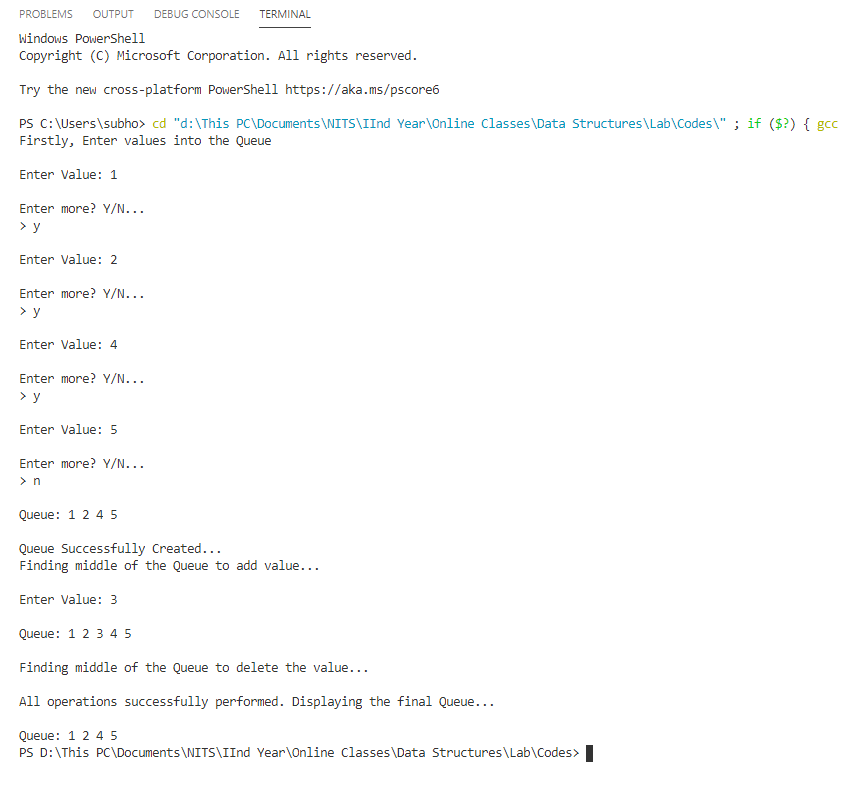
    printf ("\nAll operations successfully performed. Displaying the final Queue...\n");

    TRAVERSE ();

    return 0;

}

**OUTPUT:**

****

/\*

**Q. 2. Lab Question:** **Write a program in C to implement stack using queue.**

\*/

#include<stdio.h>

#include<stdlib.h>

#define Max 1e5

struct stack

{

    int front, rear, q1['!'], q2['!'];

}Stack;

void push (int num)

{

    if(Stack.rear == Max-1)

    {

        printf("\nSTACK OVERFLOW\n");

        return;

    }

    if(Stack.front == -1)

        Stack.front = 0;

    Stack.q1 [++(Stack.rear)] = num;

}

void pop()

{

    if(Stack.front == -1)

    {

        printf("\nSTACK UNDERFLOW\n");

        return;

    }

    else if (Stack.front == Stack.rear)

    {

        Stack.front=Stack.rear = -1;

        return;

    }

    int i = -1;

    while(Stack.front < Stack.rear)

    {

        Stack.q2 [++i] = Stack.q1 [Stack.front++];

    }

    Stack.front = 0;

    Stack.rear = -1;

    for(int j = 0; j <= i; ++j)

    {

        Stack.q1 [++(Stack.rear)] = Stack.q2 [j];

    }

}

void display()

{

    if (Stack.front == -1)

    {

        printf("\nSTACK UNDERFLOW\n");

        return;

    }

    for(int i = Stack.rear; i >= Stack.front; --i)

        printf("%d ",Stack.q1 [i]);

}

void top()

{

    if(Stack.front == -1)

    {

        printf("\nSTACK UNDERFLOW\n");

        return;

    }

    printf("\n%d\n", Stack.q1 [Stack.rear]);

}

int main()

{

    Stack.front=Stack.rear=-1;

    int data, option;

    do

    {

        printf ("\nMENU\n");

        printf ("1. Enter into Stack\n");

        printf ("2. Delete from Stack\n");

        printf ("3. Top of the Stack\n");

        printf ("4. Exit\n> ");

        scanf("%d",&option);

        switch (option)

        {

            case 1: printf ("\nEnter Value: ");

                    scanf("%d",&data);

                    push (data);

                    printf ("Added...\n");

                    break;

            case 2: pop ();

                    printf ("Deleted...\n");

                    break;

            case 3: top ();

                    break;

            default:

                    break;

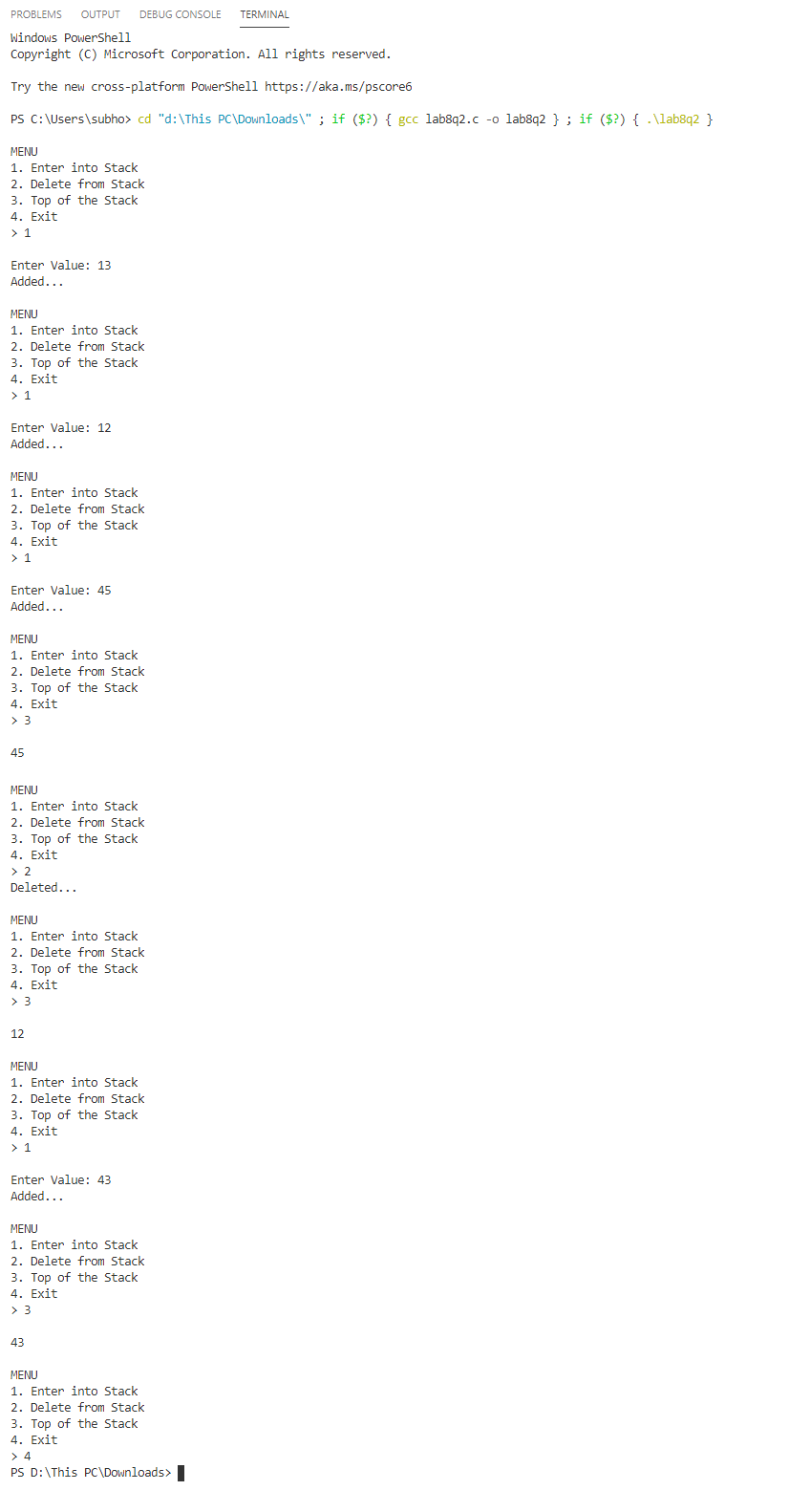
        }

    }while (option < 4);

    return 0;

}

**OUTPUT:**

****