#include<stdio.h>

#include<stdlib.h>

#define MAX\_SIZE 3

struct node

{

int data;

struct node \*next;

}\*newnode,\*front,\*rear,\*temp;

int count = 0;

void enQueue(int a){

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

newnode->data = a;

if (count == MAX\_SIZE) //checking the full condition

{

printf("queue full..");

}

else if (front == NULL)

{

front = newnode;

rear = newnode;

newnode->next = NULL;

count++; //to check whether the stack has reached to the MAX\_SIZE

printf("node enQueued successfully..");

}

else{

newnode->next = front;

front = newnode;

count++;

printf("node enQueued successfully..");

}

}

void deQueue(){

if (front == NULL)

{

printf("nothing here to delete...");

}

temp = front;

if (front == rear)

{

front = NULL;

rear = NULL;

}

while (temp->next != rear)

{

temp = temp->next;

}

temp->next = NULL;

free(rear);

rear = temp;

printf("node deQueued succesfully");

}

void traverse(){

if (front == NULL)

{

printf("nothing here to display..");

}

else{

temp = front;

while (temp->next!=NULL)

{

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

temp = temp->next;

}

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

}

}

void main(){

int ch,data;

do{

printf("\n1.enQueue\n2.deQueue\n3.Traverse\n");

scanf("%d",&ch);

switch (ch)

{

case 1:

printf("enter the number : ");

scanf("%d",&data);

enQueue(data);

break;

case 2:

deQueue();

break;

case 3:

traverse();

break;

default:

break;

}

printf("\ndo you want to continue (0/1) : ");

scanf("%d",&ch);

}while(ch<=3);

}

OUTPUT

1.enQueue

2.deQueue

3.Traverse

1

enter the number : 1

node enQueued successfully..

do you want to continue (0/1) : 1

1.enQueue

2.deQueue

3.Traverse

1

enter the number : 2

node enQueued successfully..

do you want to continue (0/1) : 1

1.enQueue

2.deQueue

3.Traverse

1

enter the number : 3

node enQueued successfully..

do you want to continue (0/1) : 1

1.enQueue

2.deQueue

3.Traverse

3

3 -> 2 -> 1

do you want to continue (0/1) : 1

1.enQueue

2.deQueue

3.Traverse

2

node deQueued succesfully

do you want to continue (0/1) : 1

1.enQueue

2.deQueue

3.Traverse

3

3 -> 2

do you want to continue (0/1) : 0