#include<stdio.h>

#include<stdlib.h>

// Declaration of struct type node

struct node

{

int data;

struct node \*next;

};

struct node \*front=-1;

struct node \*rear=-1;

// function to insert the element in the Queue

void enqueue(int x)

{

struct node \*newnode; // declaration of pointer of struct node type.

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

newnode->data=x;

newnode->next=0;

if(rear==-1) // checking whether the Queue is empty or not.

{

front=rear=newnode;

rear->next=front;

}

else

{

rear->next=newnode;

rear=newnode;

rear->next=front;

}

}

// function to delete the element from the queue

void dequeue()

{

struct node \*temp; // declaration of pointer of node type

temp=front;

if((front==-1)&&(rear==-1)) // checking whether the queue is empty or not

{

printf("\nQueue is empty");

}

else if(front==rear) // checking whether the single element is left in the queue

{

front=rear=-1;

free(temp);

}

else

{

front=front->next;

rear->next=front;

free(temp);

}

}

// function to get the front of the queue

int peek()

{

if((front==-1) &&(rear==-1))

{

printf("\nQueue is empty");

}

else

{

printf("\nThe front element is %d", front->data);

}

}

// function to display all the elements of the queue

void display()

{

struct node \*temp;

temp=front;

printf("\n The elements in a Queue are : ");

if((front==-1) && (rear==-1))

{

printf("Queue is empty");

}

else

{

while(temp->next!=front)

{

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

temp=temp->next;

}

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

}

}

void main()

{

enqueue(34);

enqueue(10);

enqueue(23);

display();

dequeue();

peek();

}