```c
# include<stdio.h>
# include<stdlib.h>
struct node
{
 int data;
 struct node*prev;
 struct node*next;
};
struct node* createnode()
{
int data;
struct node* newnode =NULL;
newnode = (struct node*) malloc(sizeof(struct node));
if(newnode == NULL)
{
 printf("Memory not allocated\n");
 return NULL;
}
else
 {
  printf("Enter Data :=>");
  scanf("%d",&data);
  newnode->data = data;
  newnode->next = NULL;
  newnode->prev  = NULL;
  return newnode;
 }
}
int IS_Full()
{
 struct node* newnode = (struct node*)malloc(sizeof(struct node));
 if(newnode == NULL)
 {
  printf("Queue is Full\n");
  return 0;
 }
 else
 {
  printf("There is space for elements\n");
  return 1;
 }
}
int IS_Empty(struct node**head)
{
 struct node* front = *head;
 if(front==NULL)
 {
  printf("Queue is Empty\n");
  return 0;
 }
 else
 {
  printf("There are elements in queue\n");
 return 1;
}
}
void Insert(struct node **front,struct node**rear)
{
 if(IS_Full())
 {
 if(*rear == NULL && *front == NULL)
 {
  struct node * newnode = createnode();
  *front = newnode;
  *rear = newnode;
 }
```

```c
 else
 {struct node * newnode = createnode();
  (*rear)->prev = newnode;
  newnode->next = *rear;
  *rear = newnode;
 }
}
}
void Delete(struct node ** front,struct node ** rear)
{
  if(IS_Empty(front))
  {
     if((*front)->data == (*rear)->data)
   {
    printf("%d is deleted\n",(*front)->data);
     *front = *rear =NULL;

   }
   else
   {
    printf("%d is deleted\n",(*front)->data);
     *front = (*front)->prev;
   }
  }

}
void Display(struct node * front)
{
 while(front!=NULL)
 {
  printf("%d ",front->data);
  front = front->prev;
 }
 printf("\n");
}

void main()
{
 struct node * front = NULL;
 struct node * rear  = NULL;
 int choice;
 do
 {
  printf("1.check full\n2.check
empty\n3.Insert\n4.Delete\n5.Display\n0.Exit\n*******************\nEnter your
choice\n");
  scanf("%d",&choice);
  switch(choice)
 {
   case 1: IS_Full();
           break;
   case 2: IS_Empty(&front);
           break;
   case 3: Insert(&front,&rear);
           break;
   case 4: Delete(&front,&rear);
           break;
   case 5: Display(front);
           break;
 }
 }while (choice!=0);
}
```