for enqueue

int q[5],size;

front=0,rear=-1;

void enqueue(int x)

{

if(rear==size-1)

cout<<"queue is full";

else{

q][++rear]=x;

}

->libraries will not check for the emptiness it gives run time error so we have to check the condition

for dequeue

void dequeue()

{

if(rear==-1)

cout<<"queue is empty";

else

{

if(front==rear){

front=0;

rear=-1;

}

else

{

front++

}

}

->to make linked list a queue we should allow the insertion at tail end and deletion at head end

Circular queue :we use this because in queue when we increment front there is some space that we are not using it when we are deleting element

To check the circular queue whether it is full or not use following conditions:

1)front=0&rear=size-1

(or)

2)front=rear+1

Make front to 0 and for rear make it rear=(rear+1)%size;

And make cq[rear]=x;

For deletion check front==-1 make front=(front+1)%size;

Void enqueue(int x){

If((front==0&&rear=size-1)||front==rear+1))

Cout<<”full cq”;

Else

{

Rear=(rear+1)%size;

Cq[rear]=x;

If(front==-1)

Front=0

}}

Void dequeue()

{

If(front==-1)

Cout<<”empty”;

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

Else{

Front=(front+1)%size;

}

Void display(){

For(i=front;i!=rear;i=(i+1)%size) { cout<<cq[i];

}

Cout<<Cq[rear];

}

Int main()

{

Menu driven program

Int cq[5],size=5,front=-1,rear=-1;

While(1)

{

Cout<<1.enqueue 2.dequeue 3.display 4.exit;

Int ch=x;cin>>ch;

Switch(ch){

Case 1:cin>>x;enqueue(x);break;

Case 2:dequeue();break;

Case 3:display();break;

Case 4: retun 0;

Default:

Cout<<”Invalid choice”;

}}