**AIM:**

Pizza parlor accepting maximum 10 order .Orderare served on first come first serve basis .order once placed can’t be cancel .write c++ to simulate the system using circular Queue

**Objective:**

To Learn implementation of circular queue to and other operations which are performed on circular queue in order to simulate pizza parlor system.

**Theory :**

Circular Queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle. It is also called **‘Ring Buffer’**

* **Front:** Get the front item from queue.
* **Rear:** Get the last item from queue.
* **enQueue(value)**This function is used to insert an element into the circular queue. In a circular queue, the new element is always inserted at Rear position.
* **deQueue()** This function is used to delete an element from the circular queue. In a circular queue, the element is always deleted from front position.

**Algorithm:**

Operations on Circular Queue:

* **Front:** Get the front item from queue.
* **Rear:** Get the last item from queue.
* **enQueue(value)**This function is used to insert an element into the circular queue. In a circular queue, the new element is always inserted at Rear position.

**Steps:**

* 1. Check whether queue is Full – Check ((rear == SIZE-1 && front == 0) || (rear == front-1)).
  2. If it is full then display Queue is full. If queue is not full then, check if (rear == SIZE – 1 && front != 0) if it is true then set rear=0 and insert element.
* **deQueue()** This function is used to delete an element from the circular queue. In a circular queue, the element is always deleted from front position.

**Steps:**

* 1. Check whether queue is Empty means check (front==-1).
  2. If it is empty then display Queue is empty. If queue is not empty then step 3
  3. Check if (front==rear) if it is true then set front=rear= -1 else check if (front==size-1), if it is true then set front=0 and return the element.

**Source code:**

#include<iostream>

#include<cstdlib>

using namespace std;

class pizza

{

int front,rear,q[5];

public:

pizza()

{

front=-1;

rear=-1;

}

int isfull()

{

if((front==0&&rear==4)||front==rear+1)

{

return 1;

}

else

{

return 0;

}

}

int isempty()

{

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

{

return 1;

}

else

{

return 0;

}

}

void add()

{

if(isfull()==0)

{

cout<<"\n Enter the Pizza ID: ";

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

{

front=0;

rear=0;

cin>>q[rear];

}

else

{

rear=(rear+1)%5;

cin>>q[rear];

}

char c;

cout<<" Do you want to add another order ? ";

cin>>c;

if(c=='y'||c=='Y')

add();

}

else

{

cout<<"\n Orders are full ";

}

}

void serve()

{

if(isempty()==0)

{

if(front==rear)

{

cout<<"\n Order served is : "<<q[front];

front=-1;

rear=-1;

}

else

{

cout<<"\n Order served is : "<<q[front];

front=(front+1)%5;

}

}

else

{

cout<<"\n Orders are empty ";

}

}

void display()

{

if(isempty()==0)

{

for(int i=front;i!=rear;i=(i+1)%5)

{

cout<<q[i]<<" <- ";

}

cout<<q[rear];

}

else

{

cout<<"\n Orders are empty";

}

}

void check()

{

int ch;

cout<<"\n\nPIZZA PARLOUR !\n\n";

cout<<"\n 1. Add a Pizza in Queue \n 2. Display the Orders \n 3. Serve a pizza \n 4. Exit \n Enter your choice : ";

cin>>ch;

switch(ch)

{

case 1:

add();

break;

case 2:

display();

break;

case 3:

serve();

break;

case 4:

exit(0);

default:

cout<<" Invalid choice ";

check();

}

char ch1;

cout<<"\n Do you want to continue? ";

cin>>ch1;

if(ch1=='y'||ch1=='Y')

{check();}

}

};

int main()

{

pizza p1;

p1.check();

return 0;

}

**Output:**

PIZZA PARLOUR !

1. Add a Pizza in Queue

2. Display the Orders

3. Serve a pizza

4. Exit

Enter your choice : 1

Enter the Pizza ID: 1

Do you want to add another order ? y

Enter the Pizza ID: 2

Do you want to add another order ? y

Enter the Pizza ID: 3

Do you want to add another order ? n

Do you want to continue? y

PIZZA PARLOUR !

1. Add a Pizza in Queue

2. Display the Orders

3. Serve a pizza

4. Exit

Enter your choice : 2

1 <- 2 <- 3

Do you want to continue? y

PIZZA PARLOUR !

1. Add a Pizza in Queue

2. Display the Orders

3. Serve a pizza

4. Exit

Enter your choice : 3

Order served is : 1

Do you want to continue? y

PIZZA PARLOUR !

1. Add a Pizza in Queue

2. Display the Orders

3. Serve a pizza

4. Exit

Enter your choice : 2

2 <- 3

Do you want to continue? n

--------------------------------

Process exited after 36.42 seconds with return value 0

Press any key to continue . . .

**Conclusion:**

Circular queue is successfully implemented to simulate pizza parlour system.