**Assignment Number: 8**

**Subject: Data Structure and Algorithm**

**Name: Shrirang Mhalgi**

**Roll No.:222006**

**Class: S.E**

**Division: B**

**Batch: B1**

**Title/Problem Statement**

Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed can not be cancelled. Write C++ program to simulate the system using circular queue using array.

**CODE**

#include<iostream>

using namespace std;

#define max 30

class pizza

{

int front,rear,queue[max],count;

public:

pizza(int no)

{

front=rear=-1;

count=0;

}

int add(int,int);

void display(int);

int del(int);

};

void pizza::display(int size)

{

int start,last;

start=front;

last=rear;

while(start!=last)

{

cout<<queue[start]<<"--->";

start=(start+1)%size;

}

cout<<queue[last]<<"--->";

cout<<"NULL\n";

}

int pizza::add(int order,int size)

{

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

{

cout<<"\nCANNOT PLACE MORE ORDERS!!!";

return 0;

}

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

{

front++;

queue[++rear]=order;

return 1;

}

else if(rear==size-1 && front!=0)

{

rear=0;

queue[rear]=order;

return 1;

}

else

{

queue[++rear]=order;

return 1;

}

}

int pizza::del(int size)

{

if(front==-1)

{

cout<<"\nNO PENDING ORDERS TO BE SERVED!!!";

return 0;

}

else if(front==rear)

{

cout<<queue[front];

front=rear=-1;

cout<<"\nORDER PLACED!!!";

return 0;

}

else if(front==size-1)

{

cout<<queue[front];

front=0;

return 1;

}

else

{

cout<<queue[front++];

return 1;

}

}

int main()

{

int max1,choice,val,flag=0;

char ch;

cout<<"\nENTER MAMXIMUM ORDERS : ";

cin>>max1;

pizza ob(max1);

do

{

cout<<"\n1. PLACE ORDER";

cout<<"\n2. SERVE ORDER";

cout<<"\n\nCHOICE : ";

cin>>choice;

switch(choice)

{

case 1: do

{

cout<<"\n\nSELECT ORDER :-";

cout<<"\n1.VEG PIZZA";

cout<<"\n2.NON VEG PIZZA";

cout<<"\nCHOICE : ";

cin>>choice;

if(choice!=1 && choice!=2)

{

flag=1;

cout<<"\nWRONG CHOICE!!! ENTER AGAIN...";

}

else

flag=0;

}while(flag==1);

val=ob.add(choice,max1);

if(val==1)

{

cout<<"\nORDER PLACED!!!";

cout<<"\nLIST OF ORDERS :-\n";

ob.display(max1);

}

break;

case 2: cout<<"\nORDER TO BE SERVED : ";

val=ob.del(max1);

if(val==1)

{

cout<<"\nORDER SERVED!!!\n";

cout<<"\nLIST OF REMAINING ORDERS :-\n";

ob.display(max1);

}

break;

default: cout<<"\nWRONG CHOICE!!!";

}

cout<<"\n\nWANT TO CONTINUE (y/n) : ";

cin>>ch;

}while(ch=='y'||ch=='Y');

return 0;

}

/\*

OUTPUT

SELECT ORDER :-

1.VEG PIZZA

2.NON VEG PIZZA

CHOICE : 1

CANNOT PLACE MORE ORDERS!!!

WANT TO CONTINUE (y/n) : y

1. PLACE ORDER

2. SERVE ORDER

CHOICE : 2

ORDER TO BE SERVED : 1

ORDER SERVED!!!

LIST OF REMAINING ORDERS :-

1--->1--->NULL

WANT TO CONTINUE (y/n) : y

1. PLACE ORDER

2. SERVE ORDER

CHOICE : 2

ORDER TO BE SERVED : 1

ORDER SERVED!!!

LIST OF REMAINING ORDERS :-

1--->NULL

WANT TO CONTINUE (y/n) : n

\*/