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

***CODE :-***

#include <iostream>

using namespace std;

class PizzaParlor

{

private:

int \*queue;

int front, rear, size, capacity;

public:

PizzaParlor(int maxOrders)

{

capacity = maxOrders;

queue = new int[capacity];

front = rear = -1;

size = 0;

}

bool isEmpty()

{

return size == 0;

}

bool isFull()

{

return size == capacity;

}

void placeOrder(int orderID)

{

if (isFull())

{

cout << "Sorry, the pizza parlor is at full capacity. Cannot accept more orders.\n";

} else {

if (front == -1)

{

front = 0;

}

rear = (rear + 1) % capacity;

queue[rear] = orderID;

size++;

cout << "Order " << orderID << " has been placed successfully.\n";

}

}

void serveOrder()

{

if (isEmpty())

{

cout << "No orders to serve. The pizza parlor is empty.\n";

} else

{

int servedOrder = queue[front];

cout << "Serving order " << servedOrder << ".\n";

front = (front + 1) % capacity;

size--;

if (size == 0)

{

front = rear = -1;

}

}

}

void displayOrders()

{

if (isEmpty())

{

cout << "No orders in the queue.\n";

} else

{

cout << "Current Orders: ";

int i = front;

for (int count = 0; count < size; count++)

{

cout << queue[i] << " ";

i = (i + 1) % capacity;

}

cout << endl;

}

}

~PizzaParlor()

{

delete[] queue;

}

};

int main()

{

int maxOrders;

cout << "Enter the maximum number of orders the pizza parlor can accept: ";

cin >> maxOrders;

PizzaParlor parlor(maxOrders);

int choice, orderID;

do

{

cout << "\n--- Pizza Parlor Menu ---\n";

cout << "1. Place an Order\n";

cout << "2. Serve an Order\n";

cout << "3. Display Current Orders\n";

cout << "4. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

cout << "Enter Order ID: ";

cin >> orderID;

parlor.placeOrder(orderID);

break;

case 2:

parlor.serveOrder();

break;

case 3:

parlor.displayOrders();

break;

case 4:

cout << "Exiting the program...\n";

break;

default:

cout << "Invalid choice! Please try again.\n";

break;

}

} while (choice != 4);

return 0;

}

***OUTPUT :-***

