ASSIGNMENT NO. 13 (Group E)

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.

**Name : Sattyam Sagar Chavan Roll No : 73 Dept : SE B (AI&DS)**

#include <iostream> using namespace std;

const int MAX = 5; // Maximum number of orders class PizzaParlour {

int front, rear;

int orders[MAX];

public:

PizzaParlour() { front = rear = -1;

}

bool addOrder(int id); void serveOrder(); void display();

};

bool PizzaParlour::addOrder(int id) {

if (front == (rear + 1) % MAX) { // Queue is full cout << "\nCafe is Full. Please wait.\n"; return false;

}

if (rear == -1) { // Queue is empty front = rear = 0;

} else {

rear = (rear + 1) % MAX; // Circular increment

}

orders[rear] = id; return true;

}

void PizzaParlour::serveOrder() {

if (front == -1) { // Queue is empty

cout << "\nNo Orders in Cafe. [Cafe is Empty]\n"; return;

}

cout << "\nOrder No. " << orders[front] << " is processed.\n"; if (front == rear) { // Only one order was in the queue

front = rear = -1; // Reset queue

} else {

front = (front + 1) % MAX; // Circular increment

}}

void PizzaParlour::display() {

if (front == -1) { // Queue is empty cout << "\nCafe has zero orders.\n"; return;

}

cout << "\nOrder IDs: "; int i = front;

do {

cout << orders[i] << " "; i = (i + 1) % MAX;

} while (i != (rear + 1) % MAX); // Loop until all orders are displayed cout << endl;

}

void intro() {

cout << "\n Vaibhav Cafe \n";

}

int main() {

int ch, id = 0;

cout << "\nName:Sattyam Sagar Chavan ; Roll No: SEBD23273; Division: B\n\n";

PizzaParlour q;

do {

cout << "\n ";

intro();

cout << " "; cout << "\n\*\*\* Menu \*\*\*\n"; cout << "1. Accept order\n"; cout << "2. Serve order\n";

cout << "3. Display orders\n"; cout << "4. Exit\n";

cout << "Choice: "; cin >> ch;

switch (ch) { case 1:

id++;

if (q.addOrder(id)) {

cout << "Thank you for your order. Order ID is: " << id << endl;

}

break;

case 2:

q.serveOrder(); break;

case 3:

q.display(); break;

case 4:

cout << "\nThank You. KeepVisiting.”; break;

default:

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

}

} while (ch != 4);

return 0;

}