**Assignment 31**

/\*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.\*/

===========================================================================

**#include**<iostream>

**using** **namespace** std;

**#define** MAX 5

**class** pizza {

**int** order[MAX];

**int** front, rear;

**public**:

**pizza**() {

front = -1, rear = -1;

}

**int** **isfull**() {

**return** (((front == 0) && (rear == (MAX - 1)))

|| (front == (rear + 1) % MAX)) ? 1 : 0;

}

**int** **isempty**() {

**return** (front == -1) ? 1 : 0;

}

**void** **add**(**int** a) {

**if** (front == -1)

front = rear = 0;

**else**

rear = (rear + 1) % MAX;

order[rear] = a;

}

**int** **remove**() {

**int** temp = order[front];

**if** (front == rear)

front = rear = -1;

**else**

front = (front + 1) % MAX;

**return** temp;

}

**void** **display**() {

**int** i;

**if** (isempty())

cout << "\nNo orders currently\n";

**else** {

cout << "\nThe orders are : \n\n";

**for** (i = front; i != rear; i = (i + 1) % MAX)

{

cout << " " << order[i];

}

cout << " " << order[i];

}

}

};

**int** **main**() {

**int** ch;

pizza p;

**do** {

cout << "\n1. Order \n2. Remove order \n3. Display orders \n4. Exit \nEnter your choice:";

cin >> ch;

**switch** (ch) {

**case** 1:

**if** (p.isfull())

cout << " Queue is overflow";

**else** {

**int** o;

cout << "\nEnter Order number : ";

cin >> o;

p.add(o);

}

**break**;

**case** 2:

**if** (p.isempty())

cout << "\n Queue is underflow";

**else** {

p.remove();

}

**break**;

**case** 3:

p.display();

**break**;

**case** 4:

cout<<"\nExit";

}

} **while** (ch != 4);

**return** 0;

}

Output:

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:1

Enter Order number : 1

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:1

Enter Order number : 2

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:1

Enter Order number : 3

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:1

Enter Order number : 4

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:1

Enter Order number : 5

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:1

Queue is overflow

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:3

The orders are :

1 2 3 4 5

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:2

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:3

The orders are :

2 3 4 5

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:2

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:2

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:3

The orders are :

4 5

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:1

Enter Order number : 6

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice:3

The orders are :

4 5 6

1. Order

2. Remove order

3. Display orders

4. Exit

Enter your choice: