Practical No. 13 (Group E)

Name : Atharva B. Iparkar

Roll no : S211045

Class : S.E.

Div : A

Batch : A-2

Problem Statement :

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;

#define MAX 5 // You can change the value of MAX to simulate different maximum orders.

class PizzaParlor {
    int front, rear, count;
    int orders[MAX];

public:
```

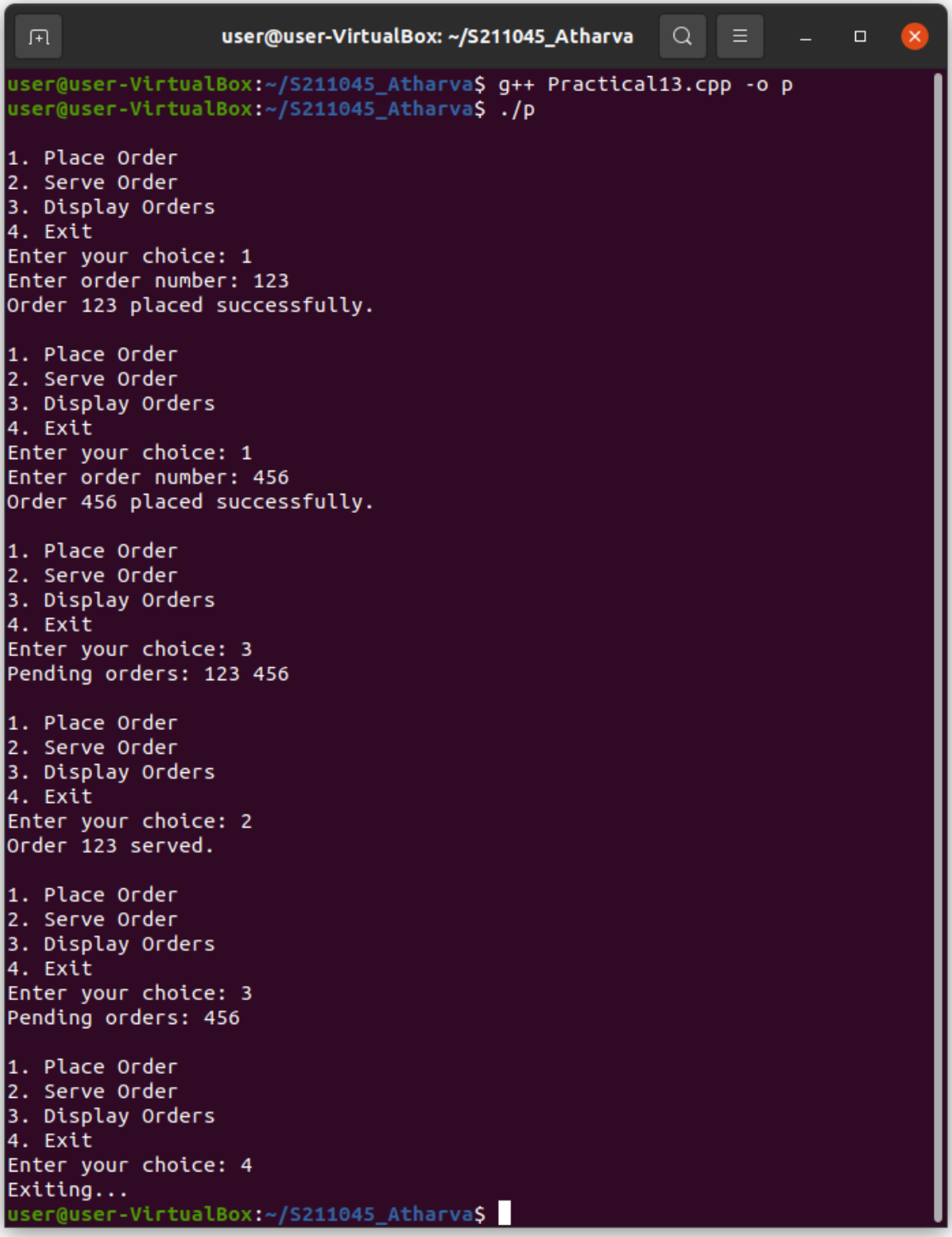
```
PizzaParlor() {
    front = -1;
    rear = -1;
    count = 0;
}

bool isFull() {
    return (count == MAX);
}

bool isEmpty() {
    return (count == 0);
}

void placeOrder(int order) { // Fixed the type from i8,nt to int
    if (isFull()) {
        cout << "Order queue is full. Cannot accept more orders.\n";
        return;
    }
    rear = (rear + 1) % MAX;
    orders[rear] = order;
    if (front == -1)
        front = 0;
    count++;
    cout << "Order " << order << " placed successfully.\n";
```

```
}

void serveOrder() {
    if (isEmpty()) {
        cout << "No orders to serve.\n";
        return;
    }
    cout << "Order " << orders[front] << " served.\n";
    front = (front + 1) % MAX;
    count--;
    if (isEmpty()) {
        front = rear = -1; // Reset front and rear when the queue becomes empty
    }
}

void displayOrders() {
    if (isEmpty()) {
        cout << "No pending orders.\n";
        return;
    }
    cout << "Pending orders: ";
    int i = front;
    for (int c = 0; c < count; c++) {
        cout << orders[i] << " ";
        i = (i + 1) % MAX;
```

```
        }
        cout << endl;
    }
};

int main() {
    PizzaParlor parlor;
    int choice, order;

    do {
        cout << "\n1. Place Order\n2. Serve Order\n3. Display Orders\n4. Exit\n";
        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter order number: ";
                cin >> order;
                parlor.placeOrder(order);
                break;
            case 2:
                parlor.serveOrder();
                break;
            case 3:
```

```
            parlor.displayOrders();
            break;
        case 4:
            cout << "Exiting...\n";
            break;
        default:
            cout << "Invalid choice. Please try again.\n";
      }
   } while (choice != 4);

   return 0;
}
```

Output :


user@user-VirtualBox: ~/S211045_Atharva
user@user-VirtualBox:~/S211045_Atharva$ g++ Practical13.cpp -o p
user@user-VirtualBox:~/S211045_Atharva$ ./p
1. Place Order
2. Serve Order
3. Display Orders
4. Exit
Enter your choice: 1
Enter order number: 123
Order 123 placed successfully.
1. Place Order
2. Serve Order
3. Display Orders
4. Exit
Enter your choice: 1
Enter order number: 456
Order 456 placed successfully.
1. Place Order
2. Serve Order
3. Display Orders
4. Exit
Enter your choice: 3
Pending orders: 123 456
1. Place Order
2. Serve Order
3. Display Orders
4. Exit
Enter your choice: 2
Order 123 served.
1. Place Order
2. Serve Order
3. Display Orders
4. Exit
Enter your choice: 3
Pending orders: 456
1. Place Order
2. Serve Order
3. Display Orders
4. Exit
Enter your choice: 4
Exiting...
user@user-VirtualBox:~/S211045_Atharva$