```
#include<iostream>
using namespace std;

// Node clas for the double-ended queue
class node {
        public:
   int customer_no;
   node *next;
   node *prev;
};

// Double-ended queue class
class deque{
public:
   node *front;
   node *rear;
   deque(){
      front = rear = NULL;
   }

   // Function to check if the queue is empty
   bool isEmpty(){
      return front == NULL;
   }

   // Function to add an element at the front of the queue (Input Restricted Deque)
   void insertFront(int customer_no){
      node *temp = new node;
      temp->customer_no = customer_no;
      temp->next = front;
```

```
    temp->prev = NULL;

    if(isEmpty()){
        rear = temp;
    }
                else{
        front->prev = temp;
    }
    front = temp;
}

// Function to add an element at the rear of the queue (Output Restricted Deque)
void insertRear(int customer_no){
    node *temp = new node;
    temp->customer_no = customer_no;
    temp->next = NULL;
    temp->prev = rear;

    if(isEmpty()){
        front = temp;
    }
                else{
        rear->next = temp;
    }
    rear = temp;
}

// Function to delete an element from the front of the queue (Output Restricted Deque)
void deleteFront(){
    if(isEmpty()){
        cout << "No customers inside the mall!" << endl;
        return;
    }

    node *temp = front;
    front = front->next;

    if(front == NULL){
        rear = NULL;
    }
                else{
        front->prev = NULL;
    }
    delete temp;
}

// Function to delete an element from the rear of the queue (Input Restricted Deque)
void deleteRear(){
    if(isEmpty()){
        cout << "No customers inside the mall!" << endl;
        return;
    }

    node *temp = rear;
    rear = rear->prev;

    if(rear == NULL){
        front = NULL;
    }
                else{
```

```
            rear->next = NULL;
        }
        delete temp;
    }

    // Function to display the elements of the queue
    void display(){
        if(isEmpty()){
            cout << "No customers inside the mall!" << endl;
            return;
        }

        node *temp = front;
        while(temp != NULL){
            cout << temp->customer_no << " ";
            temp = temp->next;
        }
        cout << endl;
    }
};

int main(){
    deque dq;
    cout << "Wellcome to Shopping Mall" << endl;
    int choice;
    int cust_no;

    do {
        cout << "\nSelect an operation: " << endl;
        cout << "1. Enter customer from front gate " << endl; // (Input Restricted Deque)
cout << "2. Enter Customer from rear gate" << endl; // (Output Restricted Deque)
cout << "3. Exiting Customer from front gate " << endl; // (Output Restricted Deque)
cout << "4. Exiting Customer from rear gate" << endl; // (Input Restricted Deque)
cout << "5. Display Customers in Mall" << endl;
cout << "6. Cancel" << endl;
cout << "Enter your choice: ";
cin >> choice;

switch(choice) {
   case 1:
      cout << "Enter the customer number entering from front gate: ";
      cin >> cust_no;
      dq.insertFront(cust_no);
      cout << "Customer entered from front gate successfully!" << endl;
      break;
   case 2:
      cout << "Enter the customer number entering from rear gate: ";
      cin >> cust_no;
      dq.insertRear(cust_no);
      cout << "Customer entered from rear gate successfully!" << endl;
      break;
   case 3:
      cout << "Exiting customer from the front gate..." << endl;
      dq.deleteFront();
      cout << "Done!"<<endl;
      break;
   case 4:
      cout << "Exiting customer from the rear gate..." << endl;
      dq.deleteRear();
      cout << "Done!" << endl;
```

```
                break;
            case 5:
                cout << "The customers inside the mall are: ";
                dq.display();
                break;
            case 6:
                cout << "Exiting the system..." << endl;
                break;
            default:
                cout << "Invalid choice! Please try again." << endl;
        }

    } while(choice != 6);

    return 0;
}
```