Assignment 8:

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.

|  |
| --- |
| /\*-----------------------------------------------------  Assignment 8:  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 <bits/stdc++.h>  using namespace std;  class Queue  {  int rear, front;  int size;  int \*arr;  public:  Queue(int s)  {  front = rear = -1;  size = s;  arr = new int[s];  }  void insert(int value);  int deleteQue();  void displayQueue();  };  void Queue::insert(int value)  {  if ((front == 0 && rear == size - 1) ||  (rear == (front - 1) % (size - 1)))  {  cout << "\nQueue is Full";  return;  }  else if (front == -1) /\* Insert First Element \*/  {  front = rear = 0;  arr[rear] = value;  }  else if (rear == size - 1 && front != 0)  {  rear = 0;  arr[rear] = value;  }  else  {  rear++;  arr[rear] = value;  }  }  int Queue::deleteQue()  {  if (front == -1)  {  cout <<"\nQueue is Empty";  return INT\_MIN;  }  int data = arr[front];  arr[front] = -1;  if (front == rear)  {  front = -1;  rear = -1;  }  else if (front == size - 1)  front = 0;  else  front++;  return data;  }  void Queue::displayQueue()  {  if (front == -1)  {  cout << "\nQueue is Empty";  return;  }  cout << "\nElements in Circular Queue are: ";  if (rear >= front)  {  for (int i = front; i <= rear; i++)  cout << arr[i] << " ";  }  else  {  for (int i = front; i < size; i++)  cout << arr[i] << " ";  for (int i = 0; i <= rear; i++)  cout << arr[i] << " ";  }  cout << endl;  }  /\* Driver of the program \*/  int main()  {  int choice;  Queue q(5);  do {  cout << "1. Insert Elements" << endl;  cout << "2. Delete Element" << endl;  cout << "3. Display Elements" << endl;  cout << "4. Exit" << endl;  cin >> choice;  switch (choice) {  case 1:  // Inserting elements in Circular Queue  for(int i = 0; i < 5; i++){  int element;  cout << "Insert an element - ";  cin >> element;  q.insert(element);  }  break;  case 2:  q.deleteQue();  break;  case 3:  q.displayQueue();  break;  case 4:  break;  default:  cout << "Wrong input"<< endl;  break;  }  }while(choice != 4);  } |