```
#include <iostream>
using namespace std;
class CircularQueue {
private:
int front, rear, size;
int* queue;
public:
CircularQueue(int s) {
size = s;
queue = new int[size];
front = rear = -1;
}
CircularQueue() {
delete[] queue;
}
void enqueue(int value) {
if ((rear + 1) % size == front) {
cout << "Queue is Full! Cannot insert " << value << endl;
return;
}
if (front == -1) {
front = rear = 0;
} else {
rear = (rear + 1) % size;
}
queue[rear] = value;
cout << "Inserted: " << value << endl;
}
void dequeue() {
if (front == -1) {
cout << "Queue is Empty! Cannot delete." << endl;
return;
}
cout << "Deleted: " << queue[front] << endl;
if (front == rear) {
front = rear = -1;
} else {
front = (front + 1) % size;
}
}
void display() {
if (front == -1) {
cout << "Queue is Empty!" << endl;
return;
}
cout << "Queue elements: ";
int i = front;
while (true) {
cout << queue[i] << " ";
if (i == rear) break;
i = (i + 1) % size;
}
cout << endl;
```

```
}
};
int main() {
int queueSize, choice, value;
cout << "Enter the size of the Circular Queue: ";
cin >> queueSize;
CircularQueue cq(queueSize);
do {
cout << "\nMenu:\n";
cout << "1. Enqueue (Insert)\n";
cout << "2. Dequeue (Delete)\n";
cout << "3. Display\n";
cout << "4. Exit\n";
cout << "Enter your choice: ";
cin >> choice;
switch (choice) {
case 1:
cout << "Enter the value to insert: ";
cin >> value;
cq.enqueue(value);
break;
case 2:
cq.dequeue();
break;
case 3:
cq.display();
break;
case 4:
cout << "Exiting...\n";
break;
default:
cout << "Invalid choice! Please try again.\n";
}
} while (choice != 4);
return 0;
}
```