```
#include <iostream>
#define MAX 5
using namespace std;

class Queue
{
   int a[MAX];
   int front, rear;

public:
   Queue()
   {
      front = -1;
      rear = -1;
   }

   bool isempty()
   {
      return (front == -1 && rear == -1);
   }

   bool isfull()
   {
      return ((rear + 1) % MAX == front);
   }

   void enqueue(int value);
```

```
    void dequeue();
    void display();
};

void Queue::enqueue(int value)
{
    if (isfull())
    {
        cout << "Queue is full. Cannot insert more elements.\n";
        return;
    }

    if (isempty())
    {
        front = 0;
        rear = 0;
    }
    else
    {
        rear = (rear + 1) % MAX;
    }

    a[rear] = value;
    cout << "Inserted element is:  " << value << "\n";
}

void Queue::dequeue()
{
```

```
    if (isempty())
    {
        cout << "Queue is empty. Cannot delete element.\n";
        return;
    }

    cout << "Deleted element is = " << a[front] << "\n";

    if (front == rear)
    {
        // Only one element was in the queue
        front = -1;
        rear = -1;
    }
    else
    {
        front = (front + 1) % MAX;
    }
}

void Queue::display()
{
    if (isempty())
    {
        cout << "Queue is empty.\n";
        return;
    }
```

```
    cout << "Queue elements are: ";
    int i = front;
    while (true)
    {
        cout << a[i] << "\t";
        if (i == rear)
            break;
        i = (i + 1) % MAX;
    }
    cout << "\n";
}

int main()
{
    Queue q;
    int value, choice;

    do
    {
        cout << "\nSelect an operation:\n1. Enqueue\n2. Dequeue\n3. Display\n4. Exit\n";
        cin >> choice;

        switch (choice)
        {
        case 1:
            cout << "Enter value to be inserted into queue: ";
            cin >> value;
            q.enqueue(value);
```

```
            break;
        case 2:
            q.dequeue();
            break;
        case 3:
            q.display();
            break;
        case 4:
            cout << "Exiting...\n";
            break;
        default:
            cout << "Wrong choice. Try again.\n";
        }
    } while (choice != 4);

    return 0;
}
```