1.)
```cpp
#include <iostream>

using namespace std;

#define SIZE 5

class Queue {
    int arr[SIZE], front, rear;
public:
    Queue() { front = rear = -1; }

    bool isEmpty() { return front == -1; }
    bool isFull() { return rear == SIZE - 1; }

    void enqueue(int x) {
        if (isFull()) cout << "Queue Full\n";
        else {
            if (front == -1) front = 0;
            arr[++rear] = x;
        }
    }

    void dequeue() {
        if (isEmpty()) cout << "Queue Empty\n";
        else {
            cout << "Deleted: " << arr[front] << endl;
            if (front == rear) front = rear = -1;
            else front++;
        }
    }

    void display() {
        if (isEmpty()) cout << "Queue Empty\n";
        else {
            for (int i = front; i <= rear; i++) cout << arr[i] << " ";
            cout << endl;
        }
    }
```

```
    void peek() {
        if (!isEmpty()) cout << "Front: " << arr[front] << endl;
        else cout << "Queue Empty\n";
    }
};

int main() {
    Queue q;
    int ch, val;
    do {
        cout << "\n1.Enqueue 2.Dequeue 3.Display 4.Peek 5.Exit\n";
        cin >> ch;
        switch (ch) {
            case 1: cin >> val; q.enqueue(val); break;
            case 2: q.dequeue(); break;
            case 3: q.display(); break;
            case 4: q.peek(); break;
        }
    } while (ch != 5);
}
```

2.)
```
#include <iostream>

using namespace std;

#define SIZE 5

class CQueue {
    int arr[SIZE], front, rear;
public:
    CQueue() { front = rear = -1; }

    bool isEmpty() { return front == -1; }
    bool isFull() { return (rear + 1) % SIZE == front; }

    void enqueue(int x) {
        if (isFull()) cout << "Queue Full\n";
        else {
            if (front == -1) front = 0;
            rear = (rear + 1) % SIZE;
            arr[rear] = x;
        }
    }
```

```
    void dequeue() {
        if (isEmpty()) cout << "Queue Empty\n";
        else {
            cout << "Deleted: " << arr[front] << endl;
            if (front == rear) front = rear = -1;
            else front = (front + 1) % SIZE;
        }
    }

    void display() {
        if (isEmpty()) cout << "Queue Empty\n";
        else {
            int i = front;
            while (true) {
                cout << arr[i] << " ";
                if (i == rear) break;
                i = (i + 1) % SIZE;
            }
            cout << endl;
        }
    }

    void peek() {
        if (!isEmpty()) cout << "Front: " << arr[front] << endl;
        else cout << "Queue Empty\n";
    }
};

int main() {
    CQueue q;
    int ch, val;
    do {
        cout << "\n1.Enqueue 2.Dequeue 3.Display 4.Peek 5.Exit\n";
        cin >> ch;
        switch (ch) {
            case 1: cin >> val; q.enqueue(val); break;
            case 2: q.dequeue(); break;
            case 3: q.display(); break;
            case 4: q.peek(); break;
        }
    } while (ch != 5);
}
```

3.)
```cpp
#include <iostream>

#include <queue>
using namespace std;

int main() {
    queue<int> q;
    int arr[] = {4,7,11,20,5,9};
    for (int x : arr) q.push(x);

    int n = q.size();
    queue<int> first;
    for (int i = 0; i < n/2; i++) {
        first.push(q.front());
        q.pop();
    }

    while (!first.empty()) {
        q.push(first.front()); first.pop();
        q.push(q.front()); q.pop();
    }

    while (!q.empty()) {
        cout << q.front() << " ";
        q.pop();
    }
}
```

4.)
```cpp
#include <iostream>

#include <queue>
using namespace std;

int main() {
    string s;
    cout << "Enter string: ";
    cin >> s;

    queue<char> q;
    int freq[256] = {0};

    for (char c : s) {
        freq[c]++;
        q.push(c);
```

```
        while (!q.empty() && freq[q.front()] > 1)
            q.pop();

        if (q.empty()) cout << -1 << " ";
        else cout << q.front() << " ";
    }
}
```

5.(a)
```
#include <iostream>

#include <queue>
using namespace std;

class Stack {
    queue<int> q1, q2;
public:
    void push(int x) {
        q2.push(x);
        while (!q1.empty()) {
            q2.push(q1.front()); q1.pop();
        }
        swap(q1, q2);
    }

    void pop() {
        if (!q1.empty()) q1.pop();
    }

    void top() {
        if (!q1.empty()) cout << "Top: " << q1.front() << endl;
    }
};

int main() {
    Stack s;
    s.push(10); s.push(20); s.push(30);
    s.top();
    s.pop();
    s.top();
}
```

5(b)
```cpp
#include <iostream>

#include <queue>
using namespace std;

class Stack {
    queue<int> q;
public:
    void push(int x) {
        q.push(x);
        for (int i = 0; i < q.size()-1; i++) {
            q.push(q.front());
            q.pop();
        }
    }

    void pop() {
        if (!q.empty()) q.pop();
    }

    void top() {
        if (!q.empty()) cout << "Top: " << q.front() << endl;
    }
};

int main() {
    Stack s;
    s.push(5); s.push(10); s.push(15);
    s.top();
    s.pop();
    s.top();
}
```