伪代码：

Enqueue：

若队列已满，返回出错；

否则插入数到尾指针处，尾指针后移一位；

Dequeue：

若队列已空，返回出错

否则删除尾指针前一位，尾指针向前移一位

代码：

#include <iostream>

using namespace std;

#define N 5

class Queue {

private:

int Que[N];

int front, rear, tag;

public:

Queue() {

front = 0; rear = 0; tag = 0;

}

int Enqueue(Queue& Q, int n);

int Dequeue(Queue& Q, int& n);

friend ostream& operator << (ostream& o, const Queue& Q);

};

int Queue::Enqueue(Queue& Q, int n) {

if (Q.front == Q.rear && Q.tag == 1) {

cout << "Enqueue False" << endl;

return 0;

}

if (Q.front == Q.rear && Q.tag == 0) {

Q.tag = -1;

Q.Que[rear] = n;

Q.rear = (Q.rear + 1) % N;

if (Q.front == Q.rear && Q.tag == -1)Q.tag = 1;

}

else {

Q.Que[rear] = n;

Q.rear = (Q.rear + 1) % N;

if (Q.front == Q.rear && Q.tag == -1)Q.tag = 1;

}

return 1;

}

int Queue::Dequeue(Queue& Q, int& n) {

if (Q.front == Q.rear && Q.tag == 0) {

cout << "Dequeue False" << endl;

return 0;

}

if (Q.front == Q.rear && Q.tag == 1) {

Q.tag = -1;

n = Q.Que[front];

Q.front = (Q.front + 1) % N;

if (Q.front == Q.rear && Q.tag == -1)tag = 0;

}

else {

n = Q.Que[front];

Q.front = (Q.front + 1) % N;

if (Q.front == Q.rear && Q.tag == -1)tag = 0;

}

return 1;

}

ostream& operator << (ostream& o, const Queue& Q)

{

if (Q.tag == -1) {

for (int i = Q.front; i % 5 != Q.rear; i++)

o << Q.Que[i % N] << ' ';

}

else if (Q.tag == 0) {

o << "This Queue is empty.";

}

else if (Q.tag == 1) {

for (int i = Q.front; i < Q.front + N; i++)

o << Q.Que[i % N] << ' ';

}

return o;

}

int main()

{

Queue Q;

Q.Enqueue(Q, 1);

Q.Enqueue(Q, 2);

Q.Enqueue(Q, 3);

Q.Enqueue(Q, 4);

cout << "Queue is: " << Q << endl;

Q.Enqueue(Q, 5);

Q.Enqueue(Q, 6);

cout << "Queue is: " << Q << endl;

int n = 0;

if (Q.Dequeue(Q, n))cout << n << endl;

if (Q.Dequeue(Q, n))cout << n << endl;

if (Q.Dequeue(Q, n))cout << n << endl;

if (Q.Dequeue(Q, n))cout << n << endl;

cout <<"Queue is: " << Q << endl;

if (Q.Dequeue(Q, n))cout << n << endl;

if (Q.Dequeue(Q, n))cout << n << endl;

cout << "Queue is: " << Q << endl;

}

结果：

