#include<iostream>

#include<string>

using namespace std;

struct Customer {

string CustomerName;

int ArrivalTime;

int ServiceTime;

int FinishTime;

};

struct FCFSQueue {

Customer Customerlist[100];

int length;

};

bool IsEmpty(FCFSQueue);

int GetLength(FCFSQueue);

void Enqueue(FCFSQueue&, Customer);

void Dequeue(FCFSQueue &, Customer);

int main()

{

FCFSQueue myQueue;

myQueue.length = 0;

Customer newCus[100];

newCus[0].CustomerName = "Tom";

newCus[0].ArrivalTime = 1;

newCus[0].ServiceTime = 0;

newCus[0].FinishTime = 0;

Enqueue(myQueue, newCus[0]);

newCus[1].CustomerName = "Bob";

newCus[1].ArrivalTime = 3;

newCus[1].ServiceTime = 0;

newCus[1].FinishTime = 0;

Enqueue(myQueue, newCus[1]);

newCus[2].CustomerName = "Tim";

newCus[2].ArrivalTime = 3;

newCus[2].ServiceTime = 0;

newCus[2].FinishTime = 0;

Enqueue(myQueue, newCus[2]);

for (int i = 0; i < myQueue.length; i++) {

cout << newCus[i].CustomerName << endl;

}

Dequeue(myQueue,newCus[0]);

Dequeue(myQueue, newCus[1]);

Dequeue(myQueue, newCus[2]);

system("pause");

return 0;

}

bool IsEmpty(FCFSQueue queue)

{

if (queue.length == 0)

{

return true;

}

else

return false;

}

int GetLength(FCFSQueue queue)

{

return queue.length;

}

void Enqueue(FCFSQueue&queue, Customer cus)

{

if (queue.length <= 100) {

queue.Customerlist[queue.length] = cus;

queue.length++;

}

else

cout << "the queue is full" << endl;

}

void Dequeue(FCFSQueue & queue, Customer cus) {

if (!IsEmpty(queue)) {

cout<< cus.CustomerName <<" is leaving" << endl;

queue.length--;

}

else

cout << "the queue is empty" << endl;

}