class Token{

private int stall = M;//车位

//private int location = 0;//所在节点编号（0到N-1）

private boolean state = true;// true为权标能传递给其他节点

private boolean[] gate = new boolean[N];// true 表示编号i出口开放

//返回true表示能关闭当前节点

public boolean close(){

for(int n = 0; n < N; n++){

if(n==节点编号){

continue;

}

if(gate[n]){

return true;

}  
 }

return false;

}

}

进车：

①：等待权标

②：如果stall不为0，则减一，让车进入，输出yes

出车：

①：等待权标

②：stall加一，让车出去，输出yes

开出口：

①：等待权标

②：gate[本节点编号]设为true，输出OK

关出口：

①：等待权标

②：调用close()

③：如果返回true则将gate[本节点编号]置false，输出OK

若返回false则输出Last Gate

在每个节点设置一个private static boolean status =true;//true表示该节点不需要权标，当出现上述四种情况时，节点会将status置为false，当权标传到它时,将权标中的state置为false

权标转移规则：

若gate[(当前节点编号+1) mod N] == true;权标传递给（节点编号+1）节点，若gate[(当前节点编号+1) mod N] == false;则判断gate[((当前节点编号+1)+1) mod N]，若为true，则传递给他，若为false，重复该操作。