National Computer Rank Examination

全国计算机等级考试专用辅导丛书

全国计算机等级考试专用辅导教程

二级 Visual FoxPro

—2012版—

希赛教育等考学院 主编

紧扣最新考试大纲，透彻精讲大纲规定考点
突出重点与难点，深入分析例题，讲练结合
提供最新真题解析，摸清考试规律，掌握实考难度

全面实用权威

问希赛教育等考学院（www.csaidk.com）可获惊喜大礼！

- ◆海量模拟试题在线测试
- ◆模拟测试软件免费下载
- ◆配套学习资料倾情奉送
- ◆众考生与教师在线交流

希赛教育 WWW.EDUCITY.CN
電子工業出版社 PUBLISHING HOUSE OF ELECTRONICS INDUSTRY http://www.phei.com.cn

National Computer Rank Examination

全国计算机等级考试专用辅导丛书

全国计算机等级考试
专用辅导教程

二级
Visual FoxPro
——2012版——

希赛教育等考学院 主编

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书由希赛教育等考学院组织编写，作为全国计算机等级考试二级 Visual FoxPro 的辅导和培训指定教程。内容紧扣教育部考试中心新推出的考试大纲，通过对历年试题进行科学分析、研究、总结、提炼而成。

本书基于最新的考试大纲和历年试题，内容紧扣大纲，全面实用。全书内容涵盖了考试大纲规定的所有知识点，对考试大纲规定的内容有重点地进行了细化和深化。阅读本书，就相当于阅读了一本详细的、带有知识注释的考试大纲。准备考试的人员可通过阅读本书掌握考试大纲规定的知识，掌握考试重点和难点，熟悉内容的分布。

图书在版编目（CIP）数据

全国计算机等级考试专用辅导教程：2012 版. 二级 Visual FoxPro / 希赛教育等考学院主编.
北京：电子工业出版社，2012.1
（全国计算机等级考试专用辅导丛书）
ISBN 978-7-121-15486-7

Ⅰ. ①全… Ⅱ. ①希… Ⅲ. ①电子计算机－水平考试－自学参考资料②关系数据库－数据库管理系统，Visual FoxPro－水平考试－自学参考资料 Ⅳ. ①TP3

中国版本图书馆 CIP 数据核字（2011）第 259164 号

策划编辑：牛　勇
责任编辑：李云静
特约编辑：赵树刚
印　　刷：三河市鑫金马印装有限公司
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：860×1092　1/16　印张：22.5　字数：720 千字
印　　次：2012 年 1 月第 1 次印刷
印　　数：4000 册　定价：43.80 元

凡所购买的电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

全国计算机等级考试（NCRE）由教育部考试中心主办，面向社会，用于考查非计算机专业人员的计算机应用知识与能力。考试客观、公正，得到了社会的广泛认可。

本书根据全国计算机等级考试二级 Visual FoxPro 语言的考试大纲编写而成，本书在组织和写作上，倾注了作者的许多精力和心血，相信对提高考生的通过率，有效地完成“考试过关”有所帮助。考生可通过阅读本书，迅速掌握考试所涉及的知识点，全面进行梳理和系统学习考试大纲中的内容。

作者权威，阵容强大

希赛教育（www.educity.cn）专业从事人才培养、教育产品开发、教育图书出版，在职业教育方面具有极高的权威性。特别是在在线教育方面，在国内名列前茅，希赛教育的远程教育模式得到了国家教育部门的认可和推广。

希赛教育等考学院是国内知名的进行计算机等级考试在线教育的大型教育机构，在该领域取得了很好的效果。组织大纲制订者和阅卷组成员编写了考试辅导教材近 20 本，内容涵盖了计算机等级考试的一级、二级、三级和四级的主要级别。组织权威专家和辅导名师录制了考试培训视频教程，对历年考试进行了跟踪研究和比较研究，编写了权威的全真模拟试题。希赛教育的计算机等级考试培训采取统一教材、统一视频、统一认证教师的形式，采取线下培训与线上辅导相结合的方式，确保学员在通过考试的前提下能真正学到有用的知识。

本书由希赛教育等考学院组织编写，参加编写的人员来自大学教学一线和企业研发团队，具有丰富的教学和辅导经验，对等级考试有深入的研究，具有极强的应试技巧、理论知识、实践经验和责任心。参加编写工作的有孙鸿飞、武慧娟、施游、胡钊源、张友生、桂阳、陈勇军、王勇、何玉云、左水林、谢顺。

在线测试，心中有数

上学吧在线测试平台（www.shangxueba.com）为考生准备了在线测试，其中有数十套全真模拟试题和考前密卷，考生可选择任何一套进行测试。测试完毕，系统自动判卷，立即给出分数。

对于考生做错的地方，系统会自动记忆，待考生第二次参加测试时，可选择“试题复习”。这样，系统就会自动把考生原来做错的试题显示出来，供考生重新测试，以加强记忆。

如此，读者可利用上学吧在线测试平台的在线测试系统检查自己的实际水平，加强考前训练，做到心中有数，考试不慌。

诸多帮助，诚挚致谢

在本书出版之际，要特别感谢教育部考试中心计算机等级考试办公室的命题专家，编者在本书中引用了部分考试原题，使本书能够尽量方便读者阅读。本书在编写过程中，参考了许多相关的文献和书籍，编者在此对这些参考文献的作者表示感谢。

感谢电子工业出版社的牛勇老师，他在本书的策划、选题的申报、写作大纲的确定，以及编辑、出版等方面，付出了辛勤的劳动和智慧，给予我们很多的支持和帮助。

感谢参加希赛教育计算机等级考试辅导和培训的学员，正是他们的想法汇成了本书的源动力，他们的意见使本书更加贴近读者。

由于编者水平有限，且本书涉及的内容很广，书中难免存在错漏和不妥之处，编者诚恳地期望各位专家和读者不吝指正和帮助，对此，我们将十分感激。

互动讨论，专家答疑

希赛教育等考学院（www.csaidk.com）是中国大型的计算机等级考试在线教育网站，该网站论坛是国内人气很旺的计算机等级考试社区。希赛教育等考学院拥有强大的师资队伍，为读者提供全程的答疑服务，在线回答读者的提问。

有关本书的意见反馈和咨询，读者可在希赛教育等考学院论坛“等级考试教材”板块中的“希赛教育等考学院”栏目上与作者进行交流。

希赛教育等考学院

目　录

第 1 章　算法和数据结构

本章主要介绍算法、线性表、栈和队列、二叉树的概念，介绍几种常见的排序技术。结合计算机等级考试二级 VFP 考试大纲的要求，具体如表 1-1 所示。

表 1-1　考试要求

考试知识点	重要性
算法、线性表基本概念	★
栈和队列	★★★
树和二叉树	★★★★★
查找技术	★
排序技术	★★★★

1.1　算法与数据结构概述

本节的主要考点集中在算法与数据结构的基本概念上，包括算法的基本特征、复杂度，以及数据结构的表示等。

1.1.1　算法的概念

算法（Algorithm）是一系列解决问题的清晰指令，也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。

1. 算法的基本特征

（1）有穷性：一个算法必须在有限的时间内做完。

（2）确定性：算法中的第一个步骤都必须是有明确定义的。

（3）拥有足够的情报：有一个或多个输出、有零个或多个输入。

（4）可行性：即考虑到实际的条件能够达到一个满意的结果。

2. 算法的基本要素

（1）算法中对数据的运算和操作：每个算法实际上是按解题要求从环境能进行的所有操作中选择合适的操作所组成的一组指令序列。

计算机可以执行的基本操作是以指令的形式描述的。一个计算机系统能执行的所有指令的集合，

称为该计算机系统的指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令，所组成的指令序列。在一般的计算机系统中，基本的运算和操作有以下 4 类。

- 算术运算：主要包括加、减、乘、除等运算。
- 逻辑运算：主要包括“与”、“或”、“非”等运算。
- 关系运算：主要包括“大于”、“小于”、“等于”、“不等于”等运算。
- 数据传输：主要包括赋值、输入、输出等操作。

（2）算法的控制结构：一个算法的功能不仅取决于所选用的操作，而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

3. 算法设计的基本方法

计算机算法不同于人工处理的方法，下面是工程上常用的几种算法设计，在实际应用时，各种方法之间往往存在着一定的联系。

（1）递推法。递推法是利用问题本身所具有的一种递推关系求问题的解的一种方法。它把问题分成若干步，找出相邻几步的关系，从而达到目的。

（2）递归。递归指的是一个过程：函数不断引用自身，直到引用的对象已知。

（3）穷举搜索法。穷举搜索法是对可能是解的众多候选解按某种顺序进行逐一枚举和检验，并从中找出那些符合要求的候选解作为问题的解。

（4）贪婪法。贪婪法是一种不追求最优解，只希望得到较为满意解的方法。贪婪法一般可以快速得到满意的解，因为它省去了为找最优解要穷尽所有可能而必须耗费的大量时间。贪婪法常以当前情况为基础作最优选择，而不考虑各种可能的整体情况，所以贪婪法不要回溯。

（5）分治法。分治法是把一个复杂的问题分成两个或更多的相同或相似的子问题，再把子问题分成更小的子问题，直到最后子问题可以简单地直接求解，原问题的解即子问题解的合并。

（6）动态规划法。动态规划是一种在数学和计算机科学中使用的，用于求解包含重叠子问题的最优化问题的方法。其基本思想是，将原问题分解为相似的子问题，在求解的过程中通过子问题的解求出原问题的解。动态规划的思想是多种算法的基础，被广泛应用于计算机科学和工程领域。

（7）迭代法。迭代法是在数值分析中通过从一个初始估计出发寻找一系列近似解来解决问题（一般是解方程或者方程组）的过程，为实现这一过程所使用的方法统称迭代法。

4. 良好的算法设计的要求

一个良好的算法应达到如下目标：

（1）正确性（Correctness）。算法的计算结果必须是正确的。

（2）可读性（Readability）。可读性好有助于用户对算法的理解，不易理解的程序易于隐藏较多错误，难以调试和修改。

（3）健壮性（Robustness）。当输入数据非法时，算法也能适当地做出反应或进行处理，而不会产生莫名其妙的输出结果。

（4）效率与低存储量需求。效率指的是在程序执行时，对于同一个问题如果有多个算法可以解决，执行时间短的算法效率高；存储量需求指在算法执行过程中所需要的最大存储空间。

1.1.2 算法的复杂度

算法复杂度分为空间复杂度和时间复杂度。

1. 算法的时间复杂度

算法的时间复杂度，是指执行算法所需要的计算工作量。同一个算法用不同的语言实现，或者用不同的编译程序进行编译，或者在不同的计算机上运行，效率均不同。

2. 算法的空间复杂度

算法的空间复杂度是指执行这个算法所需要的内存空间。一个算法所占用的存储空间包括算法程序所占的空间、输入的初始数据所占的存储空间，以及算法执行中所需要的额外空间。

【例题 1】算法的空间复杂度是指_____。（2009 年 9 月）

A．算法在执行过程中所需要的计算机存储空间

B．算法所处理的数据量

C．算法程序中的语句或指令条数

D．算法在执行过程中所需要的临时工作单元数

【例题分析】

由以上定义得知，此题选 A。

1.1.3　数据结构的定义

数据结构（Data Structure）是指相互之间存在一种或多种特定关系的数据元素的集合。

数据（Data）是对客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

数据元素（Data Element）是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。

一般情况下，在具有相同特征的数据元素集合中，各个数据元素之间存在某种关系（即连续），这种关系反映了该集合中的数据元素所固有的一种结构。在数据处理领域中，通常把数据元素之间这种固有的关系简单地用前后件关系（或直接前驱与直接后继关系）来描述。

一般来说，数据元素之间的任何关系都可以用前后件关系来描述。

1. 数据的逻辑结构

数据结构是指反映数据元素之间关系的数据元素集合的表示。通俗地说，数据结构是指带有结构的数据元素的集合。用 D 表示数据元素的集合，用 R 来表示数据元素之间的前后件的关系。即一个数据结构可以表示为 $B=(D,R)$，其中 B 表示数据结构。这就是一个二元关系的表示方式。所谓结构实际上是指数据元素之间的前后件关系。

一个数据结构应包含以下两方面信息：

（1）表示数据元素的信息。

（2）表示各数据元素之间的前后件关系。

数据的逻辑结构是对数据元素之间的逻辑关系的描述。它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。

2. 数据的存储结构

数据的逻辑结构在计算机存储空间中的存放形式，称为数据的存储结构（也称为数据的物理结构）。

由于数据元素在计算机存储空间中的位置关系可能与逻辑关系不同，因此，为了表示存放在计算机存储空间中的各数据元素之间的逻辑关系（即前后件关系），在数据的存储结构中，不仅要存放各数据元素的信息，还需要存放各数据元素之间的前后件关系的信息。

一种数据的逻辑结构根据需要可以表示成多种存储结构，常用的结构有顺序、链接、索引等存储结构，采用不同的存储结构，其数据处理的效率是不同的。因此，在进行数据处理时，选择合适的存储结构是很重要的。

1.1.4 数据结构的表示

数据结构除了用二元关系表示外，还可以直观地用图形表示。

在数据结构的图形表示中，对于数据集合 D 中的每一个数据元素用中间标有元素值的方框表示，一般称之为数据节点，并简称为节点；为了进一步表示各数据元素之间的前后件关系，对于关系 R 中的每一个二元组，用一条有向线段从前件节点指向后件节点。

在数据结构中，没有前件的节点称为根节点；没有后件的节点称为终端节点（也称为叶子节点）。

一个数据结构中的节点可能是动态变化的。根据需要或在处理过程中，可以在一个数据结构中增加一个新节点（称为插入运算），也可以删除数据结构中的某个节点（称为删除运算）。插入与删除是对数据结构的两种基本运算。除此之外，对数据结构的运算还有查找、分类、合并、分解、复制和修改等。

1.1.5 线性结构与非线性结构

根据数据结构中各数据元素之间前后件关系的复杂程度，一般将数据结构分为两大类：线性结构与非线性结构。

线性结构满足如下条件：

（1）有且只有一个根节点。

（2）每一个节点最多有一个前件，也最多有一个后件。

如果一个数据结构不是线性结构，称之为非线性结构。如果一个数据结构中一个数据元素都没有，则称该数据结构为空。线性结构与非线性结构都可以是空的数据结构。对于空的数据结构，如果对该数据结构的运算是按线性结构的规则来处理的，则属于线性结构；否则属于非线性结构。

1.2 线性表

本节主要考查线性表的基本概念，以及线性表的顺序存储方式。

1.2.1 线性表概述

线性表是一种常用的数据结构。

在实际应用中，线性表都是以栈、队列、字符串、数组等特殊线性表的形式来使用的。由于这些特殊线性表都具有各自的特性，因此，掌握这些特殊线性表的特性，对于数据运算的可靠性和提高操作效率都是至关重要的。

线性表是一个线性结构，它是一个含有 n（$n \geqslant 0$）个节点的有限序列。对于其中的节点，有且仅有一个开始节点，没有前驱但有一个后继节点；有且仅有一个终端节点，没有后继但有一个前驱节点。其他的节点都有且仅有一个前驱和一个后继节点。一般地，一个线性表可以表示成一个线性序列：k_1，k_2，…，k_n，其中 k_1 是开始节点，k_n 是终端节点。

线性结构的基本特征如下：

（1）集合中必存在唯一的一个“第一元素”。

（2）集合中必存在唯一的一个“最后元素”。

（3）除最后一个元素之外，均有唯一的后继（后件）。

（4）除第一个元素之外，均有唯一的前驱（前件）。

由 n（$n\geqslant 0$）个数据元素（节点）a_1，a_2，…，a_n 组成的有限序列。

数据元素的个数 n 定义为表的长度。

当 n=0 时称为空表。

常常将非空的线性表（n>0）记作：$(a_1,a_2,\cdots,a_n)$。

1.2.2　线性表的顺序存储

线性表的顺序存储指的是用一组地址连续的存储单元依次存储线性表的数据元素。

1. 线性表的顺序存储基本概念

线性表的顺序存储结构具备如下两个基本特征：

（1）线性表中的所有元素所占的存储空间是连续的。

（2）线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

假设线性表的每个元素需要占用 k 个存储单元，并以所占的存储位置 ADR(a_i+1)和第 i 个数据元素的存储位置 ADR(a_i)之间满足下列关系：ADR(a_i+1)=ADR(a_i)+k。

线性表第 i 个元素 a_i 的存储位置为：ADR(a_i)=ADR(a_1)+(i−1)×k。

公式中 ADR(a_1)是线性表的第一个数据元素的存储位置，通常称为线性表的起始位置或基址。

在 VFP 语言中，通常定义一个一维数组来表示线性表的顺序存储空间。

a[0]	a[1]	a[2]	a[3]	a[4]

图 1-1　顺序存储

在用一维数组存放线性表时，该一维数组的长度通常要定义得比线性表的实际长度大一些，以便对线性表进行各种运算，特别是插入运算。

在线性表的顺序存储结构下，可以对线性表做以下运算：

（1）在线性表的指定位置处加入一个新的元素（即线性表的插入）。

（2）在线性表中删除指定的元素（即线性表的删除）。

（3）在线性表中查找某个（或某些）特定的元素（即线性表的查找）。

（4）对线性表中的元素进行排序（即线性表的排序）。

（5）将一个线性表分解成多个线性表（即线性表的分解）。

（6）将多个线性表合并成一个线性表（即线性表的合并）。

（7）复制一个线性表（即线性表的复制）。

（8）逆转一个线性表（即线性表的逆转）等。

2. 顺序表的基本操作

顺序表的基本操作包括插入运算和删除运算。

1）顺序表的插入运算

线性表的插入运算是指在表的第 i（$1 \leqslant i \leqslant n+1$）个位置上，插入一个新节点 x，使长度为 n 的线性表：

$$(a_1, \cdots, a_{i-1}, a_i, \cdots, a_n)$$

变成长度为 $n+1$ 的线性表：$(a_1, \cdots, a_{i-1}, x, a_i \cdots, a_n)$

现在分析算法的复杂度。设它的值为 n。该算法的时间主要花费在循环节点后移语句上，该语句的执行次数（即移动节点的次数）是 $n-i+1$。由此可看出，所需移动节点的次数不仅依赖于表的长度，而且还与插入位置有关。

当 $i=n+1$ 时，由于循环变量的终值大于初值，节点后移语句将不进行；这是最好情况，其时间复杂度为 $O(1)$。

当 $i=1$ 时，节点后移语句将循环执行 n 次，需移动表中所有节点，这是最坏情况，其时间复杂度为 $O(n)$。

综合以上的情况，得出算法的平均时间复杂度为 $O(n)$。

2）顺序表的删除运算

线性表的删除运算是指将表的第 i（$1 \leqslant i \leqslant n$）个节点删除，使长度为 n 的线性表$(a_1, \cdots, a_{i-1}, a_i, a_{i+1}, \cdots, a_n)$变成长度为 $n-1$ 的线性表$(a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n)$。

该算法的时间分析与插入算法相似，节点的移动次数也是由表长 n 和位置 i 决定的。

若 $i=n$，则由于循环变量的初值大于终值，前移语句将不执行，无须移动节点。

若 $i=1$，则前移语句将循环执行 $n-1$ 次，需移动表中除开始节点外的所有节点。在这两种情况下，算法的时间复杂度分别为 $O(1)$和 $O(n)$。

综合以上的情况得出，在顺序表上进行删除运算，平均要移动表中约一半的节点，平均时间复杂度也是 $O(n)$。

1.3 栈和队列

栈和队列都是特殊的线性表，其定义符合线性表的定义，其操作也类似于线性表的操作，只不过增加了一些限定而已。

1.3.1 栈的定义与操作

栈（Stack）是一种特殊的线性表。栈是只能在表的一端进行插入和删除运算的线性表，通常称插入、删除的这一端为栈顶（Top），另一端为栈底（Bottom），如图 1-2 所示。当表中没有元素时称为空栈。栈顶元素总是后插入的元素，从而也是最先被删除的元素；栈底元素总是最先被插入的元素，从而也是最后才能被删除的元素。

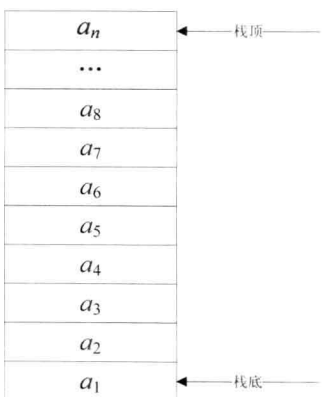

图 1-2 栈

假设栈 $S=(a_1, a_2, a_3, \cdots, a_n)$，则 a_1 称为栈底元素，a_n 为栈顶元素。栈中元素按 a_1，a_2，a_3，…，a_n 的次序进栈，退栈的第一个元素应为栈顶元素。换句话说，栈的修改是按后进先出的原则进行的。因此，栈称为先进后出表（First In Last Out，FILO），或

“后进先出”表（Last In First Out，LIFO）。

栈的操作主要有入栈运算、退栈运算（出栈）和读栈顶元素。

（1）入栈运算：入栈运算是指在栈顶位置插入一个新元素。首先将栈顶指针加一（即 top 加 1），然后将元素插入到栈顶指针指向的位置。当栈顶指针已经指向存储空间的最后一个位置时，说明栈空间已满，不可能再进行入栈操作。这种情况称为栈“上溢”错误。

（2）退栈运算：退栈是指取出栈顶元素并赋给一个指定的变量。首先将栈顶元素（栈顶指针指向的元素）赋给一个指定的变量，然后将栈顶指针减 1（即 Top 减 1）。当栈顶指针为“。”时，说明栈空，不可进行退栈操作。这种情况称为栈的“下溢”错误。

（3）读栈顶元素：读栈顶元素是指将栈顶元素赋给一个指定的变量。这个运算不删除栈顶元素，只是将它赋给一个变量，因此栈顶指针不会改变。当栈顶指针为 0 时，说明栈空，读不到栈顶元素。

1.3.2　队列的定义与操作

队列（Queue）是只允许在一端删除，在另一端插入的顺序表，允许删除的一端叫做队头（Front），允许插入的一端叫做队尾（Rear），如图 1-3 所示。

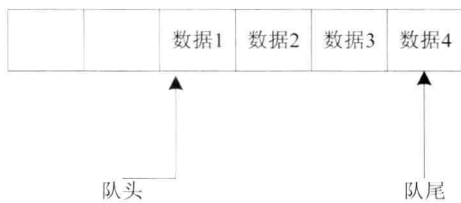

图 1-3　队列

1. 队列的运算

当队列中没有元素时称为空队列。在空队列中依次加入元素 $a_1, a_2, \cdots, a_n$ 之后，a_1 是队头元素，a_n 是队尾元素。显然退出队列的次序也只能是 $a_1, a_2, \cdots, a_n$，也就是说队列的修改是按先进先出的原则进行的。因此队列也称为先进先出（First In First Out，FIFO）的线性表，或后进后出（Last In Last Out，LILO）的线性表。

（1）入队操作。往队列队尾插入一个元素称为入队运算。

（2）出队操作。从队列的队头删除一个元素称为出队运算。

2. 循环队列的运算

所谓循环队列，就是将队列存储空间的最后一个位置绕到第一个位置，形成逻辑上的环状空间。

在循环队列中，用队尾指针 rear 指向队列中的队尾元素，用队头指针 front 指向排头元素的前一个位置。因此，从排头指针 front 指向的后一个位置到队尾指针 rear 指向的位置之间所有的元素均为队列中的元素。

在循环队列中进行出队、入队操作时，头尾指针仍要加 1，向前移动。只不过当头尾指针指向向量上界（Queuesize-1）时，其加 1 操作的结果是指向向量的下界 0。

由于入队时尾指针向前追赶头指针，出队时头指针向前追赶尾指针，故队空和队满时头尾指针均相等。因此，我们无法通过 front=rear 来判断队列是“空”还是“满”。在实际使用循环队列时，为了能区分队列满还是队列空，通常还需增加一个标志 s，定义如下：当 s=0 时表示队列空；当 s=1