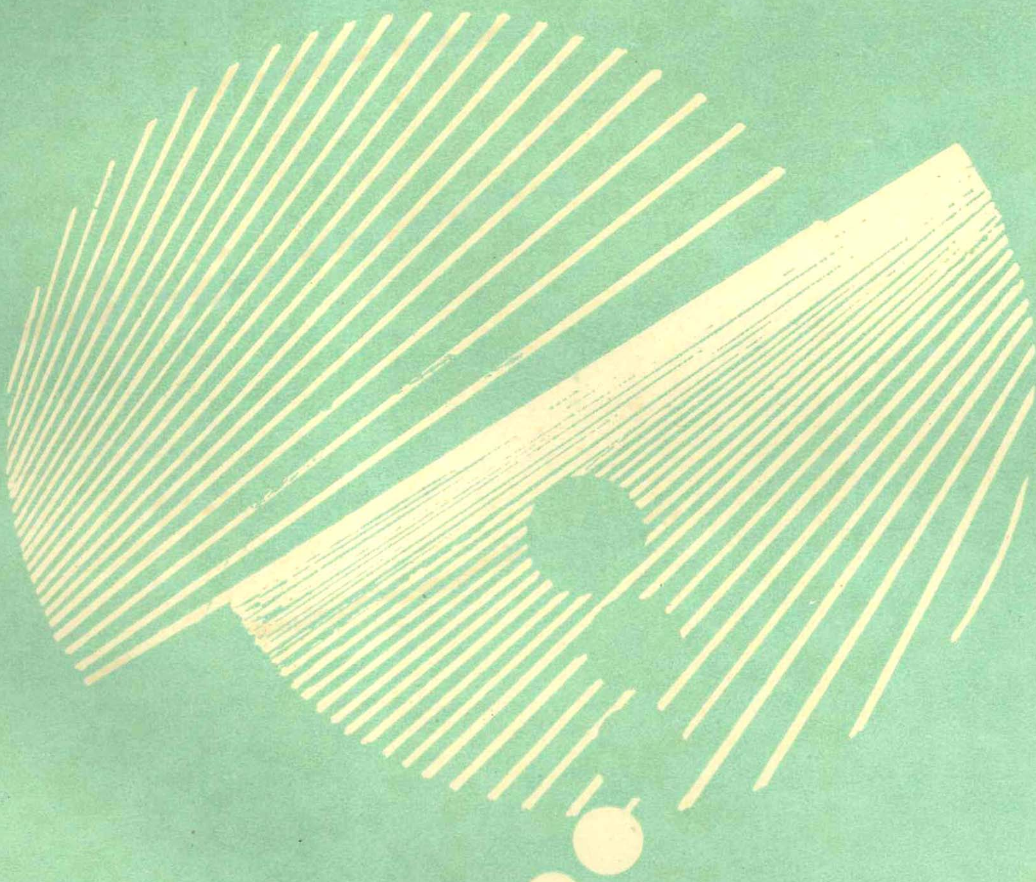
计算机基础

与FORTRAN 77程序设计

主编　宋顺林



东南大学出版社

计算机基础

与FORTRAN77程序设计

主编　宋顺林

主审　杨文显

东南大学出版社

内 容 提 要

本书是根据普通高校非计算机专业计算机基础知识和应用能力等级考试大纲，为本科生编写的计算机基础知识和FORTRAN77程序设计教学用书。

全书共14章，主要介绍计算机基础知识，WPS文字处理系统和FORTRAN77程序设计语言，论述了结构化程序设计的思想和方法，提供了大量的应用程序举例和国内高校常用的IBM-PC、VAX11／780上机操作指南，也列出了英、汉字编码供读者查阅。

本书亦可作为大专、函授和培训教材，以及科技人员的自学用书。

责任编辑 张 克 吴明新

责任校对 吴明新

计算机基础与FORTRAN77程序设计

宋顺林 主编

*

东南大学出版社出版发行

(南京四牌楼2号 邮编210018)

江苏省新华书店经销 江苏理工大学印刷厂印刷

*

开本787×1092毫米 1／16 印张 20.25 字数 486 千字

1995年9月第1版 1995年9月第1次印刷

印数：1-3000册

ISBN 7—81050—031—7／TP・4

定价：17.2 元

(凡因印装质量问题，可直接向承印厂调换)

前　　言

从 20 世纪下半叶开始，世界迈人信息时代，计算机技术的发展在其中起着重要作用．计算机技术是一门内容非常广泛，发展极其迅速，交叉渗透性强，应用领域涉及社会各方面的技术．我国计算机科学起步较晚，与发达国家相比还存在相当的差距．计算机技术的不足和高科技人才计算机应用能力的限制，使得我国某些高新技术未能发挥最佳效益．因此，迫切需要重视和加强培养普通高校非计算机专业学生计算机基础知识和应用能力，这对造就大批高质量的计算机应用人才，使我国在科技领域接近和赶上发达国家水平，具有极其深远的意义．

为了深化大专院校非计算机专业的计算机教学改革，提高计算机教学水平和广大学生计算机应用能力，主动适应社会主义建设的需要，提高人才素质，经江苏省教育委员会研究，决定从 1993 年底开始进行江苏省普通高校非计算机专业学生计算机基础知识和应用能力等级考试的试点工作，然后逐步推广．针对不同专业对计算机应用知识和应用能力的不同要求，把考试划分成三个等级：一级、二级和三级．

根据考试指导委员会颁布的有关第二等级考试的设定目标、考试范围和 FORTRAN77 程序设计语言的考试要求，我们编写了《计算机基础与 FORTRAN77 程序设计》一书，通过课堂教学和上机实习，使学生具有计算机的基础知识和使用 FORTRAN77 语言编制程序及上机调试的能力，顺利通过第二等级的考试．

计算机基础知识是作为计算机应用人才所必备的计算机硬件知识、计算机软件知识和计算机操作使用的基础知识．WPS 是一个集编辑、制表、排版、打印为一体的文字处理系统，掌握文字处理技术和实现办公自动化是计算机应用的一个重要方面，它将在我们以后的工作中起到越来越大的作用．

FORTRAN77 语言内容丰富、功能较强，是当今世界上广泛应用于科技与工程计算的一种高级程序设计语言．在一定操作系统的支持下，还可引人汉字，以适应国内开发各种汉字软件的需要．书中强调了数据文件的使用和字符处理功能，把应用扩展到非数值计算的各个领域中去．

本书编写过程中力求做到概念清楚、深入浅出、突出重点，引导初学者逐步掌握程序设计的基本方法．书中所提供的程序举例都已在机器上调试通过，可供读者在学习和工作时参考．

本书共分 14 章，第 1，4，6 章和附录由宋顺林编写；第 2，3，14 章由毕建良编写；第 12 章由夏义勇编写；第 5，13 章由邢桂芬编写；第 7，8 章和部分附录由刘政群编写；第 9 章由林庆编写；第 10，11 章由李维益编写．全书由宋顺林主编，杨文显副教授认真仔细地审阅了书稿，并提出了许多宝贵意见．此外，本书在编写出版过程中得到江苏理工大学教务处、成人教育学院、杂志社、计算机科学系的关心、指导和大力支持，在此，我们一并向他们表示衷心的感谢．

由于编写时间仓促及编者水平有限，书中错误及不妥之处在所难免，敬请读者不吝赐教。

编　者

1994年10月于江苏理工大学

目 录

1 计算机基础知识

1.1 概　述

计算机的发明是20世纪人类最伟大的科学成就之一．它的出现和应用，给社会带来了深刻的变化，有力地推动了科学技术的发展，促进了现代化管理水平的提高．如今，计算机应用日益普及，它不仅广泛应用于科学计算、过程控制和数据处理，而且已渗透到办公、教育、家庭等许许多多领域，把人类带入了信息时代．在未来的社会里，每一个有知识的人都应该学会使用计算机．计算机知识将成为新一代知识分子知识结构和智能结构的一个重要组成部分．为了使用计算机，首先必须了解计算机．

1.1.1 什么是计算机

1946年，世界上第一台计算机ENIAC是作为一种利用电子线路实现数值运算的计算工具，被研制出来的．随着计算机技术的发展，它的内涵已经发生了很大的变化，突出地表现在以下三个方面：其一，计算机不同于其他任何计算工具，如算盘、计算尺、手摇计算机或袖珍计算器等，它不仅仅是一种辅助的静止的工具，而是具有一定"智能"的自动化计算工具，故又称它为电脑．其二，尽管计算机的发明和诞生是出于"计算"的需要，但是现代计算机的功能却远远超出了当初的这个目标．它能接收各种各样的信息，对它们进行多种处理，产生满足不同需求的结果或效应，是一种自动化的工具．它在人类的社会活动中充当助手，从事人类能从事的或难以从事的某些活动．其三，计算机是一个飞速发展的、技术密集的、高科技的产物．在它的设计思想上、制造工艺上、构成材料上、理论研究上、系统构筑上以及应用技术上都经历了且正在经历着不断发展和突破的过程．与1946年世界上的第一台计算机相比，现代计算机无论在哪方面都是面貌一新的．可以预见，正在研制开发的新一代计算机将突破迄今为止的传统设计思想而竖起新设计思想的旗帜，实现"计算机是会思考的机器"的目标．

综上所述，计算机并非如它的名字标明的那样只是一种"计算"的工具，而是一种处理"信息"的工具．需要解释的是，计算机处理信息是将信息加以扩大、分配、转换和调整并"制造"出新的信息，因而更深入地说，计算机能"制造未来的知识世界"．

1.1.2 计算机的主要特点

计算机之所以有十分广泛的应用，是因为它具有任何其他计算工具不具备的众多的独有的性能特点，主要是：

1) 信息的存储和记忆性能．现代社会是一个"信息膨胀"、"知识爆炸"的社会，信息是仅次于资源和能源的第三个要素．因此，利用计算机实现对信息管理和处理的首要问题，是要求计算机具有存储大量信息和长期保存信息的能力．据现代科学证明，人脑细胞共有

130 亿个，其容量可以容纳五亿本书的信息总和，记忆力可以持续七八十年之久。虽然计算机的信息存储容量还不能与“神奇”的人脑相比，但是随着计算机技术的发展，其存储容量越来越大。60 年代的计算机只能存储和记忆几千个符号，70 年代为几万或几十万个，而到 80 年代，一台计算机可存储的信息已达几亿个符号，而且还在不断地增长之中。可以预料，随着对信息处理的更高要求，计算机的存储容量将会几倍乃至几百几千倍地增长。

计算机的记忆力也是惊人的。当初只能是临时性的记忆，现代计算机可以记住信息达几年甚至几十年，如果采用一定的技术，则记忆可以是“永久”性的。

2) 高速处理的性能。计算机的最大特点是高速处理信息的性能，这是人类无法达到的。最初，一台计算机每秒可以进行几千次运算，现代计算机每秒可以进行几百万次乃至几亿次运算。我国自行设计和安装的银河计算机即是亿次计算机的代表。计算机的高速处理性能使过去不能解决的大量复杂的科学问题得到解决，为揭示许多科学奥秘提供了不可缺少的有力工具。1867 年法国天文学家达姆尼为了用天体力学方法求解月球运行轨道花了 20 年时间。而后来，人们用计算机重复他的工作，只花了 20 小时，还发现了他的三个错误。

计算机的高速处理性能使过去只有理论意义的课题具有了实践应用意义。例如数值天气预报，如果用人工方法处理天气预报数据，当获得预报结果时，预报的实际时间早已过去，因而失去预报的意义而成为“后报”。现在用计算机可以在几分钟之内处理完气象资料，及时得出今后十天的准确的预报。再如防空系统，在计算机实时系统控制下，一秒钟内就可以对监视区内的一切变化作出反应。

此外，计算机的高速处理性能在情报的有效利用，提高生产率和管理水平等多方面都有巨大的作用和影响。统计资料表明，每年国内外公开发表的文献多达几百万篇以上，发行的杂志几万种。如果在这样的信息海洋中要寻找一个需要的信息，没有具有高速处理能力的计算机的帮助是不可能实现的。

3) 逻辑判断性能。计算机的逻辑判断能力也是人类无法比拟的，它是实现自动化的前提。例如，根据病人的临床情况，断定患了什么病，开什么样的处方，是一个复杂的逻辑判断过程。再如情报检索处理，根据读者要求找出相应的资料也是一个逻辑判断极强的处理。因此，计算机必须具备对数值的大小、事物特征的异同、现象的真假等等之类的判断能力。

4) 精确可靠的性能。这意味着计算机处理的结果必须是精确的、可靠的，否则计算机就不可信赖。用计算机可以算出 200 万位的 π 值，这个精确度是十分惊人的。一般而言，计算机的正常精确度可达 18 位，采用特殊的技术之后，可以达到满足人们需要的精确度。

可靠性尤为重要。任何一个不可靠的卫星轨道计算可以造成生命和财富的巨大损失；一个银行业务处理系统如果不可靠就可能产生破产的后果。一台正常运行的计算机是可靠的。在许多国家，决策人十分地信赖计算机的处理结果，而对人的处理结果总持怀疑的态度。

可靠性派生出来的另一个问题是安全性和保密性。使用单位对计算机中存储和处理的信息是十分珍惜的，视为他的“命根子”。因此计算机系统提供了帮助他们对信息加密的措

施，计算机的安全保密性问题也成为当今计算机能否广泛应用的重大问题。

5) 自动连续处理性能。计算机能实现自动化的关键在于它的自动连续处理能力，这也是与其他计算工具的根本差别之一。你只要一次把计算或处理的工作程序存储在计算机中，它就可按照这个程序自动地连续工作，最后向你提供需要的结果（一个计算公式的结果数据，一份报表等等），其间不再需要人工进行任何干预，而且这个工作程序在你需要时可以任意地再执行一次。这就是说，计算机一旦"学会"了你提供的计算或处理方法之后，再作这样计算或处理时就无需再"教"它了。

6) 广泛应用性能。今天，计算机的广泛应用已经毋庸置疑了。自 1946 年第一台计算机问世至今的 40 多年中，计算机经历了一个迅速发展的过程，计算机产品多种多样，名目繁多，令人眼花缭乱。但是，它们大都具有通用性的品格。

可以这样说，计算机是一种多才多艺的东西，你教它做什么，它就会什么；它"学会"了做什么，它就能做什么。同一台计算机，你可以用它作计算，解一个复杂的方程；也可以用它来管理帐目或仓库，甚至协助管理人员管理一个工厂；可以用它来驾驶飞机，控制一台车床的工艺生产过程，甚至控制核电站的核裂变过程；可以用它来设计一座高层建筑或一种家具；可以用它来从事教学，教学生学习数学、外语、弹钢琴；还可以和它下棋、打桥牌、做游戏。此外，它还可以从事美术、音乐创作以及你希望它要做的一切，只要你"教"给它。

总之，计算机的应用十分广泛，其应用领域还在不断拓广和延伸中。据统计，到目前为止，计算机已在大约 5000 多个不同的领域中得到成功的应用，而且新的应用领域还在与日俱增。

1.1.3 数制及其转换

计算机内部的数据都是以二进制形式存储和处理的。在日常生活和数学当中，为人们所习惯及常用的数制是十进制，其计数原则为"逢十进一"和"借一当十"。为了使读者能熟悉这种数制的转变，在此介绍数制及其转换。在进位计数制中的两个要素为：进位基数和位的权数。

1) 进位基数。在进位制中，可以用到的数码个数

例如：十进制数的基数为 10，所用的数码为 0，1，2，…，9 共十个数。十六进制数的基数为 16，所用的数码为 0，1，2，…，9，A，B，C，D，E，F 共十六个数。这里借用了六个字母来代表大小相当于十进制数 10，11，…，15 等六个数字。

二进制数的基数为 2，所用的数码为 0 和 1。R 进制数的基数为 R，所用的数码为 0，1，2，…，$R-1$，共 R 个数。

2) 位的权数。在进位记数制中，一个数每一位上的数字 1 所对应的一个固定的数值，称为这一位的权数。该位数的实际大小等于该位的权数乘上该位的数码。权数是一个幂。R 进制的数 $N=N_{n-1}N_{n-2}\cdots N_1N_0N_{-1}N_{-2}\cdots N_{-m}$，其各位的权数分别对应于 R^{n-1}，R^{n-2}，…，R^1，R^0，R^{-1}，R^{-2}，…，R^{-m}。基数为 R 的权必为 R^k（k 为整数）。同一形式的一个数，在不同的数制中，可以代表不同的值。例如，形如 1101.011 的数在二进制、十进制、十六进制下各位对应的权数如下：

进　制	从左向右各位对应的权数						
二进制	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
十进制	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}
十六进制	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}

3) R 进制数转化成十进制数

将一非十进制的数转化成为十进制数，可以将每位的数码值乘上对应各位的权数然后累加即可：

$$
\begin{aligned}
(N_{n-1}N_{n-2}\cdots N_0.N_{-1}N_{-2}\cdots N_{-m})_R &= (N_{n-1}\cdot R^{n-1}+N_{n-2}\cdot R^{n-2}\cdots \\
&+N_0\cdot R^0+N_{-1}\cdot R^{-1}+\cdots+N_{-m}\cdot R^{-m})_{10} \\
&=(\sum_{i=n-1}^{-m} N_i R^i)_{10}
\end{aligned}
$$

例如： $(1101.011)_2=2^3+2^2+2^0+2^{-2}+2^{-3}=(13.375)_{10}$

$(1101.011)_{16}=16^3+16^2+16^0+16^{-2}+16^{-3}=(4353.00414414)_{10}$

$(1101.011)_8=8^3+8^2+8^0+8^{-2}+8^{-3}=(577.0175)_{10}$

4) 十进制数转换成 R 进制数

(1) 整数的转换，按“除 R 取余”法

例如：十进制数 123 转换成二进制数、八进制数和十六进制数的过程如下：

```
2|123    1………二进制数最低位
2| 61    1
2| 30    0
2| 15    1
2|  7    1
2|  3    1
    1………………二进制数的最高位
```

即：$(123)_{10}=(1111011)_2$

```
8|123    3
8| 15    7
    1
```

即：$(123)_{10}=(173)_8$

```
16|123    11   用 B 代表
    7
```

即：$(123)_{10}=(7B)_{16}$

(2) 纯小数的转化：按“乘积取整”法

把给定的十进制小数乘以 R，所得到的整数作为 R 进制高位小数，截取整数以后的小数部分再乘 R 取整，直至小数部分为 0 或所得 R 进制的小数位已满足精度要求。

例如：十进制数 0.315 转换成二进制数，八进制数和十六进制数的过程如下：

$$\begin{array}{r} 0.315 \\ \times \qquad 2 \\ \hline \boxed{0}.630 \\ \times \qquad 2 \\ \hline \boxed{1}.260 \\ \times \qquad 2 \\ \hline \boxed{0}.520 \\ \times \qquad 2 \\ \hline \underbrace{\boxed{1}}_{\text{整数部分}}.\underbrace{040}_{\text{小数部分}} \end{array} \qquad \begin{array}{r} 0.315 \\ \times \qquad 8 \\ \hline \boxed{2}.520 \\ \times \qquad 8 \\ \hline \boxed{4}.160 \\ \times \qquad 8 \\ \hline \boxed{1}.280 \\ \times \qquad 8 \\ \hline \underbrace{\boxed{2}}_{\text{整数部分}}.\underbrace{240}_{\text{小数部分}} \end{array} \qquad \begin{array}{r} 0.315 \\ \times \qquad 16 \\ \hline \boxed{5}.040 \\ \times \qquad 16 \\ \hline \boxed{0}.640 \\ \times \qquad 16 \\ \hline \boxed{10}.240 \\ \times \qquad 16 \\ \hline \underbrace{\boxed{3}}_{\text{整数部分}}.\underbrace{840}_{\text{小数部分}} \end{array}$$

$$(0.315)_{10} = (0.0101)_2 = (0.2412)_8 = (0.50A3)_{16}$$

从上述转化过程可知，不同进制的小数有时不能绝对精确地进行相互转换，一般只转换指定位数，取近似值即可。

5) 二进制、八进制及十六进制之间数的转换

因 $8 = 2^3$ 故 1 位八进制数恰好可用 3 位二进制数完全表示。同理：$16 = 2^4$，故 1 位十六进制数，恰好可用 4 位二进制数完全表示，反之亦然。八进制与十六进制之间数的相互转换，一般借助于二进制数进行。

例：$(11011.01)_2 = (011011.010)_2 = (33.2)_8$

$= (00011011.0100)_2 = (1B.4)_{16}$

$(174.56)_8 = (001111100.101110)_2$

$= (\underbrace{111}\ \underbrace{1100}.\underbrace{1011}\underbrace{10})_2$

$= (\underbrace{0111}\ \underbrace{1100}.\underbrace{1011}\ \underbrace{1000})_2$

$= (7C.B8)_{16}$

6) BCD 码：二～十进制

BCD 码是十进制数在机器中的二进制编码。BCD 码用 4 位二进制数表示 1 位十进制数字，它具有二进制的形式，又具有十进制的特点。因 4 位二进制数共有 16 种编码，从中选出 10 种编码有多种方法，故 BCD 编码有多种形式，最常用的是“8421”码。它是用四位二进制表示一个十进制数字；从左向右为 8（$=2^3$）位，4（$=2^2$）位，2($=2^1$)位，1（$=2^0$）位，故称之为 8421 码。编码的二进制值恰好与对应的十进制数字等值（见表 1－1）。这种编码方式最自然且最易识别。另一种是“余三码”，同样是用 4 个二进制位，但编码的二进制值比原十进制值数字的值多 3（见表 1－1）。这种编码在机内处理时有许多方便之处。此外，还有许多不同的编码方法，如 5421 码、2421 码、循环码等等。

表 1-1 常用的 BCD 编码

数字	0	1	2	3	4	5	6	7	8	9
8421 码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
余 3 码	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

读者可以看出，就一一对应的码值来看，它们之间有一个等值关系。对 8421 码，十进制数字等于二进制编码值。余 3 码数字值等于编码值减去 3。但是，如果是几位十进制数字构成的数，就不能等同。如 1994 的 8421 码为 0001 1001 1001 0100 而不是 11111001010。

1.1.4 机内代码

1) 机器数与真值

每个数都有正负之分，在计算机中数的符号是用数码表示的，一般情况下用“0”表示正，用“1”表示负。通常符号位放在数的最高位，或者最高两位。例如：用 8 位二进制表示一个数，最高位为符号位，其余位为数字位。如：$(+21)_{10}=(00010101)_2$，$(-21)_{10}=(10010101)_2$ 连同一个符号位在一起作为一个数，称之为机器数；而它所代表的数值称为机器数的真值。

2) 机器数的几种表示

机器数有原码、反码、补码、移码几种形式。下面介绍它们的定义及其加减规则。所举的例子都以 8 位二进制数为例。

(1) 原码表示法 (True form)

假设 x 为纯小数，用小数点左面一位表示数的符号，则

$$[x]_{原}=\begin{cases} x & (0 \leqslant x < 1) \\ 1-x & (-1 < x \leqslant 0) \end{cases}$$

例如：8位二进制的原码：

最大小数为 0.1111111，十进制为$\frac{127}{128}$

最小小数为 1.1111111，十进制为$-\frac{127}{128}$

正小数零为 0.0000000

负小数零为 1.0000000

把小数点移到所有有效数字之后（最右端），这时的机器数代表一个整数。

最大整数为 01111111，十进制为127

最小整数为 11111111，十进制为-127

在原码表示法中，零的表示并非唯一，即有正零和负零两种表示法。所以，8 位二进制原码只能表示 $2^8-1=255$ 个数。如上所述，原码是用最高一位二进制位表示符号位，其余位表示其真值的机器数。此法表示简单、明显，但在计算机内原码的加减操作比较复杂，须先经过符号位的比较，有时还要经过绝对值比较才能求出结果的符号和数值。

(2) 反码表示法 (One's Complement)

n 位二进制小数的反码定义:

$$[x]_{反}=\begin{cases}x & (0\leqslant x<1)\\(2-2^{-n})+x & (-1<x\leqslant 0)\end{cases}$$

例如: 8 位二进制的反码:

最大小数为 0.1111111 (十进制为$\frac{127}{128}$)

最小小数为 1.0000000 (十进制为$-\frac{127}{128}$)

最大整数为 01111111 (十进制为$2^7-1=127$)

最小整数为 10000000 (十进制为$-2^7+1=-127$)

正 零 为 00000000

负 零 为 11111111

实质上，反码的最高位是符号位，正数的反码与原码相同，负数的反码符号位为“1”，其余各位对应原码求反，“0”变“1”，“1”变“0”。

反码的加减运算:

$$[x]_{反}+[y]_{反}=[x+y]_{反}$$

注意:

① $[x-y]_{反}=[x+(-y)]_{反}=[x]_{反}+[(-y)]_{反}$，符号位与其它数位一起参加运算，不加区分。② 在运算中，循环进位，即最高符号位的进位加到最低位上。运算的结果超出计算机所能表示的范围称为溢出。两个符号相同的（或不同的）数进行加法（或减法），如果结果的符号与加数（或者被减数）的符号不同，可以判断此时产生了溢出。

(3) 补码 (Two's Complement)

模二补码表示法定义:

$$[x]_{补}=\begin{cases}x & (0\leqslant x<1)\\2+x & (-1\leqslant x<0)\end{cases}$$

例如: 8 位二进制数的补码

最大小数为 0.1111111 (十进制为$\frac{127}{128}$)

最小小数为 1.0000000 (十进制为 -1)

最大整数为 01111111(十进制为 127)

最小整数为 10000000(十进制为 -128)

整数零为 00000000

注: 仅有一个正零，无负零。

实际上，正数的补码等于原码，负数的补码等于它的反码在最低位上加上 1。补码表示法的加减运算最为简单，都可变成加法运算。

补码的加减运算:

$$[x]_{补}+[y]_{补}=[x+y]_{补}$$

$$[x]_{补}-[y]_{补}=[x-y]_{补}=[x]_{补}+[(-y)]_{补}$$

(4) 移码表示法（也称增码表示法）

移码多用于表示浮点数中的阶码．如果移码 x 为 n 位，（包括一位符号位），其范围为 $-2^{n-1} \leqslant x \leqslant (2^{n-1}-1)$，则：

$$[x]_{移} = 2^{n-1} + x$$

例如：设 8 位移码，则 $n=8$

最大移码　11111111（十进制 $+127$）

最小移码　00000000（十进制 -128）

因此，8 位二进制移码可表示 2^8 个数，最小的数用全 0 表示．

移码的加减规则：

$$[x+y]_{移} = [x]_{移} + [y]_{移} + 2^{n-1}$$

$$[x-y]_{移} = [x]_{移} + [(-y)]_{移} + 2^{n-1}$$

(5) 例题

设：$x=-21$，$y=21$；

可求得：　$[x]_{原} = (10010101)_2$　$[x]_{反} = (11101010)_2$

$[x]_{补} = (11101011)_2$　$[x]_{移} = (01101011)_2$

$[y]_{原} = [y]_{反} = [y]_{补} = (00010101)_2$

$[y]_{移} = (10010101)_2$

$[x+y]_{补} = (00000000)_2$　　$[x+y]_{反} = (11111111)_2$

$$\begin{array}{rr}
 & 11101011 \\
+ & 00010101 \\
\hline
 & \boxed{1}00000000
\end{array}
\qquad
\begin{array}{rr}
 & 11101010 \\
+ & 00010101 \\
\hline
 & 11111111
\end{array}$$

$[x-y]_{补} = (11010110)_2$　　$[x-y]_{反} = (11010101)_2$

$$\begin{array}{rr}
 & 11101011 \\
- & 00010101 \\
\hline
 & 11101011 \\
+ & 11101011 \\
\hline
 & \boxed{1}11010110
\end{array}
\qquad
\begin{array}{rr}
 & 11101010 \\
- & 00010101 \\
\hline
 & 11101010 \\
+ & 11101010 \\
\hline
 & \boxed{1}11010100 \\
+ & 1 \\
\hline
 & 11010101
\end{array}$$

$[x+y]_{移} = (10000000)_2$　　$[x-y]_{移} = (01010110)_2$

$$\begin{array}{rr}
 & 01101011 \\
 & 10010101 \\
+ & 10000000 \\
\hline
 & \boxed{1}10000000
\end{array}
\qquad
\begin{array}{rr}
 & 01101011 \\
 & 01101011 \\
+ & 10000000 \\
\hline
 & \boxed{1}01010110
\end{array}$$

注意：上述运算中，最高位的进位被舍去（反码运算除外）．

3) 定点数和浮点数表示

(1) 定点小数

一个 m 位的定点小数在机器中的表示：小数点的位置固定在小数最高位之前，即在

小数最高位和符号位之间，形如:

N_s .	N_{-1}	N_{-2}	…	$N_{-(m-1)}$	N_{-m}

符号位 小数点　　　　数值部分

(2) 定点整数

一个 m 位的定点整数在机器中的表示：小数点的位置固定在数据最低位的右边。形如:

N_s	N_{m-1}	N_{m-2}	…	N_1	N_0 .

符号位　　　　数值部分　　　　小数点

带符号的定点数一般用原码、补码或反码表示。用定点数表示不带符号的整数时（零和正整数，称为无符号数)，符号位也用作数值的一部分。这时可表示 2^m 个数，范围是 0 ～2^m-1。

(3) 浮点数表示法

一个数 N 用浮点形式表示（即科学表示法)，可写成:

$$N=M\cdot R^E$$

其中:① R 为基数，一般取 2，8 和 16。一旦机器确定好基数值，就不能再改变，因此，机器中不存储基数。② E 为指数，指数在计算机内的表示称为阶码。阶码一般用定点整数的补码或移码表示。③M 表示尾数，一般用定点小数的原码表示，便于乘除运算。

尾数用于表示数的有效数字，其位数反映数据的精确度。当基数为 2，尾数最高位为 1 时，称此数为规格化的数。有些机器在存储时，常常隐其最高位（不存储)，使有效位多一位，从而使精度更高。

设浮点数（字长为 48 位）的阶码为 8 位，尾数占 40 位（不隐含最高位）都采用补码表示，基数为 2，则机器数形如:

4746　　3938

N_s	阶码	尾数

尾数符号

- 阶码表示的范围为−128 到+127;
- 尾数表示的范围为−1 到 $1-2^{-39}$;
- 所表示的数值范围为:

$$-2^{127}\sim+2^{127}\times(1-2^{-39})$$

4) 其它几种机内代码

(1) ASCⅡ码

ASCⅡ码是美国信息交换标准码（Amcrican Stardard Code for Information Interchange)，是目前计算机用于表示各种符号的最普遍的一种代码形式，用 7 位二进制编码表示一个字符。附录 5 列出了该代码表示的字母、数字、符号和一些控制代码，共有 $2^7=128$ 种。

(2) EBCDIC 码（扩展二—十进制标准交换码)

EBCDIC 码是采用一个字节（八位）二进制位编码的，它的字符与代码对应关系如