计算机基础教育丛书

True BASIC程序设计

谭浩强　张基温

清华大学出版社

True BASIC程序设计
谭浩强　张基温
清

版社

内 容 简 介

True BASIC 是由 BASIC 语言的两位创始人推出的新的 BASIC 版本，对 BASIC 语言作了重大的改进和发展。本书系统地介绍了 True BASIC 的结构化与模块化程序设计方法，同时通过大量例题介绍算法及如何编程，每章后均附有习题。本书采用了新的体系编写，从算法入手，使读者在学习之后能具有较强的算法设计和程序设计的能力，养成良好的程序设计风格。

本书可作为高等院校、中专和各类计算机培训班的教材，也可供自学参考。

(京)新登字158号

True BASIC 程序设计

谭浩强　张基温　编著

责任编辑　范素珍

☆

清华大学出版社出版

北京　清华园

中国科学院印刷厂印装

新华书店总店科技发行所发行

☆

开本：787×1092 1/16　印张：19.5　字数：462 千字

1989年7月第1版　1992年6月第4次印刷

印数：28001—38000

ISBN 7-302-00505-2/TP・179

定价：7.50元

丛书出版说明

近年来，我国的计算机应用事业迅速发展，大批科技人员、大中学生、管理人员、以及各行各业的在职人员都迫切要求学习计算机知识，他们已经认识到，计算机知识是当代知识分子的知识结构中不可缺少的重要部分。

计算机应用人才的队伍由两部分人组成：一部分是从计算机专业毕业的计算机专门人才，他们是计算机应用人才队伍中的骨干力量；另一部分是各行各业中从事计算机应用的人才，他们既熟悉本专业的业务，又掌握计算机应用的技术，人数众多，是计算机应用人才队伍的基本力量。他们掌握计算机知识的情况和应用计算机的能力在相当大程度上决定了我国计算机应用的水平。因此，在搞好计算机专业教育的同时，在广大非计算机专业中开展计算机基础教育是十分必要的。

非计算机专业中的计算机教学，无论就目的、内容、教学体系、教材、教学方法等各方面都与计算机专业有很大的不同，它以应用为目的，以应用为出发点。如果不注意这个特点，将会事倍功半。广大非计算机专业的师生、在职干部迫切希望有一套适合他们的教材，以便循序渐进地迈入计算机应用领域，并且不断地提高自己的水平。我们在前几年陆续编写了一些适合初学者使用的教材，受到广大群众的欢迎。许多读者勉励我们在此基础上进一步摸索和总结规律，为我国的广大非计算机专业人员编写一整套合适的教材。

近年来，全国许多专家、学者在这个领域作了有益的探索，写出了一批受到群众欢迎的计算机基础教育的教材。特别是全国高等学校计算机基础教育研究会作了大量的工作，在集思广益的基础上，提出了在高等学校的非计算机专业中进行计算机教育的四个层次的设想，受到广泛的注意和支持。我们认为：计算机的应用是分层次的，同样，计算机人才的培养也是分层次的；非计算机专业中各个领域的情况不同，也不能一律要求，在进行计算机教育时也应当有不同的层次。对于每一个学习计算机知识的人，还有一个由浅入深，逐步提高的过程。

我们认为，编辑出版一套全面而有层次的计算机基础教育的教材，目前不仅是十分必要的，而且是完全有条件的。在全国高等学校计算机基础教育研究会和许多同志的积极推动和清华大学出版社的大力支持下，我们决定编辑《计算机基础教育丛书》。它的对象是：高等学校非计算机专业的学生、计算机继续教育或培训班的学员、广大在职自学人员。

本丛书包括计算机科学技术的一些最基本的内容，例如计算机各种常用的高级语言、计算机软件技术基础、计算机硬件技术基础、微型计算机的原理与应用、算法与数据结构、数据库基础、计算机辅助设计基础、微机网络与应用、系统分析与设计等，形成多层次的结构，读者可以根据需要与可能选学。

本丛书的宗旨是针对广大非计算机专业的需要和特点来组织教材，敢于破除框框，从实际出发，用读者容易理解的体系和叙述方法，深入浅出、循序渐进地帮助读者更好地掌握课

程的基本内容。希望我们的丛书能在这方面闯出自己的风格，在实践中接受检验。

本丛书的作者大多数是高等学校中有较丰富教学经验的教师。但是，由于计算机科学技术的飞速发展以及我们的水平有限，丛书肯定会存在许多不足，丛书的书目和内容也应当不断发展和更新。我们热情地希望得到社会各界和广大读者的批评指正。

主编　谭浩强　林定基　刘瑞挺

1988.10

前　言

近年来，计算机的应用已在我国迅速推广，计算机的知识已成为当代科技人员和管理人员的知识结构中不可缺少的一部分。为了使用计算机，人们需要学习计算机语言。BASIC语言是一种既容易学又有广泛实用价值的计算机高级语言。前几年在我国已有近千万人学习了BASIC语言。不少人从学习BASIC语言入门，迈进了计算机应用的领域。BASIC的突出优点是充分考虑初学者的特点，它是为初学者设计的一种计算机语言，从1964年BASIC问世以来，它以极其迅速的速度在全世界推广开来。它受到千百万初学者的热烈欢迎，在我国同样也是如此。BASIC语言在计算机的推广普及中已经并将继续发挥巨大的作用。

由于BASIC语言的出现比较早，和早期出现的其它语言一样，它不是结构化的。虽然近年来不少计算机厂商提供的BASIC版本扩充了功能，提供了结构化的语句，但是由于缺乏统一的标准，形成了多种BASIC版本并存的局面，造成了BASIC的“方言”化。此外，由于BASIC采取解释执行方式，程序执行的速度较慢。

在计算机科学技术迅速发展的当今，所有的语言都在不断地发展和完善，特别是要适应结构化程序设计的要求。1984年，BASIC语言的创始人John G. Kemeny和Thomas E. Kurtz对BASIC语言作了重大的改进和发展，提出了名为True BASIC（意为“真正的BASIC”）的新的BASIC版本，它严格遵循美国国家标准BASIC的规定。True BASIC是一种理想的计算机语言，它不仅完全适应结构化与模块化程序设计的要求，而且保留了BASIC语言的优点——易学易懂，程序易编写易调试。它并不比PASCAL语言逊色，而且在某些方面优于PASCAL语言，尤其是它不仅适用于数值计算、数据处理，还有丰富的作图功能。这两位BASIC的创始人宣称，True BASIC的出现将开辟BASIC的新纪元。

True BASIC的出现的确给BASIC带来新的生命力。学习True BASIC将会使读者养成编制程序的良好风格。应当大力推广True BASIC。

为了适应发展，并且满足广大计算机初学者和爱好者的要求，我们编写了本书，系统地介绍True BASIC程序设计。本书不是简单地介绍True BASIC语法规则，而是着眼于算法，因为算法是程序设计的灵魂，语言只是实现算法的一种工具。学习语言的目的是利用语言工具进行程序设计。本书把重点放在提高编写程序的能力上。

本书在写法上，采取科学性与通俗性相结合的方式，通过大量例题深入浅出地介绍如何使用True BASIC语言编写各类程序。即使从未接触过计算机的读者，也能循序渐进地学懂本书的内容。

由于我们水平有限，难免存在错误之处，恳请专家和读者不吝指正。

谭浩强　张基温

1988.10.

目　录

第一章　计算机算法

电子计算机自1946年问世以来，以迅猛的速度发展，在短短的四十多年中，已经历了四个发展阶段，即所谓的“四代”——电子管计算机、晶体管计算机、集成电路计算机，大规模集成电路计算机，现在正在研制着第五代计算机。

计算机的出现是科学技术发展史上的一场伟大的革命，对人类社会产生了深远的影响。现在几乎每个领域都在大力开展计算机的应用。

计算机并不神秘，学会使用计算机亦非难事。计算机是按照人们的意旨进行工作的。为了进行运算，人们必须事先准备好数据和程序。所谓程序，是指用计算机语言表示的操作步骤。例如，一个BASIC程序就是用BASIC语言写出的让计算机如何进行操作的一系列指令。

为了编好程序，就必须事先整理出解题的思路，正确地拟定出每一个操作步骤。“算法”就是研究这个问题的。本章将介绍算法、计算机和程序的有关知识。

§1.1　算法与计算机

1.1.1　算法的特征

初学者往往认为计算机解题是一个不可思议的过程。其实，计算机解题的过程与操作机器、执行任务、演奏音乐、打算盘、打太极拳、炒菜、做饭等极为类似。

广义地说，做任何事情都是一个过程，这个过程可以看作为在规定条件下能够进行的基本操作所组成的序列。例如，开汽车的过程可以看作是如下一个可进行的基本操作序列：

1. 掏出钥匙；2. 打开车门；3. 坐在座位上；4. 打开电门；5. 打火；6. 加油；7. 踩离合器；……。或者说，我们在做任何事之前，都要在规定的解题环境之内，用所允许的基本操作去构造解题的步骤，只不过在做那些我们已经熟悉（习惯）的工作时，无须再有意识地考虑它罢了。例如，歌手唱歌、乐队演奏，都要为他们事先设计乐谱，乐谱就是由音符组成的序列，而音符就是音乐中所规定的基本操作的符号。同样，打太极拳的图谱、做菜的菜谱、珠算的口诀、工作计划、生产流程、治病处方、……，都代表了在不同的解题环境中，为实现某一目的所应完成的基本操作序列。只不过在不同的解题环境下，对基本操作有不同的定义。这种解题操作序列称为“算法”。这里“解题”二字是泛指解决某一问题，而不只是指“计算”。计算机科学家D.E.Knuth说：“一个算法是一个有穷规则的集合，其中的规则规定了一个解决某一特定类型问题的解答。”算法的概念源于数学，当然包括了在数学中的应用。

例 1.1.1　有三个数a,b,c,找出其中最大的数。

我们先考虑处理这类问题的思路，先将a和b两数相比，将大者放在变量max中，然后再让c与max相比，如果c>max，将c的值送到max中，最后max就是三个数中最大的数。上台将它写成以下形式：

S1：将a与b比，如果a>b，则使a的值作为max的值，否则，使b的值作为max的值。

可以写成：如a＞b，则a⇒max，否则，b⇒max。

S2：将c与max比，如c＞max，则⇒cmax。

最后max的值就是三个数中最大的数。

上面用S1,S2表示步骤的次序，在写算法时常用这种形式的标记。S是step（步）的首字母。S1代表“第一步”，S2代表“第二步”，……。

例 1.1.2 有两个变量a和b,要求将它们的值互换。例如a＝3,b＝4,互换后,a＝4,b＝3。

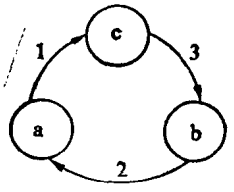

图 1.1

为了进行两个变量的值互换，须引入一个临时变量。正如二个瓶子内的液体互换需要用第三个瓶子作为过渡一样，见图1.1。其算法可表示如下：

S1：a⇒c（将a的值送给变量c）；

S2：b⇒a（将b的值送给变量a）；

S3：c⇒b（将c的值送给变量b）。

通过以上三个步骤实现了两个变量值的交换。这个方法在以后程序设计中时常会用到。

例 1.1.3 求1＋2＋3＋4＋5,即求$\sum_{n=1}^{5}n$。

先用最原始的方法进行：

S1：先进行1＋2的运算，相加的和（结果）放在变量N中（N的值为3）。

S2：将N再加3，和仍放在N中（此时N的值为6）。

S3：将N的值再加4，和仍放在N中（此时N的值为10）。

S4：将N的值再加5，和仍放在N中（此时N的值为15）。

将可算法写成如下形式：

S1：令N＝1

S2：令I＝2

S3：N与I相加，结果放在N中

S4：使I的值加1，即I＋1⇒I

S5：如果I≤5，返回S3，否则算法终止

最后N的值就是$\sum_{n=1}^{5}n$的结果。

在执行这个算法时，S3和S4两个步骤要重复执行多次。直到I＞5为止（请读者考虑S3和S4重复执行多少次？）。可以看到，这个算法是采用循环的方法来处理的。请考虑：在S5步骤中的循环继续的条件为什么是“I≤5”，如果改写为“I＜5”会出现什么情况？

显然，这个算法是比较好的。如果需要求的是$\sum_{n=1}^{100}n$，即1＋2＋…＋100，上述算法基本上可不必改，只需将S5步骤中的“若I≤5”改为“若I≤100”即可。读者可自己写出此完整的算法，并分析它的执行过程。

例 1.1.4 有十个数，找出其中最大的数。

本题的思路：如同打擂台一样，台上先站着一个人，第二个人上台与之比较，胜者留在台上，然后第三个人上台再与台上的人（胜者）比较,胜者留在台上,比完九轮之后，在台上的人便是最后胜者。

今先将输入的第一个数放在MAX中，将输入的第二个数与之相比，如第二个数大于

MAX，则它取代MAX的原值。然后第三个数再与MAX比，大者放在MAX中……，一直到比完九次为止。可以写成以下的第一种算法：

S1：输入一个数，放在MAX中。

S2：1⇒I（I用来累计比较次数）。

S3：输入一个数，放在X中。

S4：若X＞MAX，则X⇒MAX。

S5：I＋1⇒I（I的值加1）。

S6：若I≤9，返回S3继续执行。否则，停止。此时MAX的值就是十个数中最大的数。

例 1.1.5　求两个正整数m和n（m＞0，n＞0，m＞n）的最大公约数。

求两个正整数的最大公约数的算法是介绍算法时常常引用的一个典型例子。为了使读者容易理解，我们首先看看手工求M＝60和n＝33的最大公约数的过程：

S1：以n(33)除m(60)，得余数r(27)。

S2：判断r是否等于零：若r＝0，则n为解，若r≠0，则进行S3的操作（现r＝27，故进入操作S3）。

S3：以n作为新的m(33)，以r作为新的n(27)求新的m/n的余数r(6)。

S4：判断r是否为零：若r＝0，则前一个n即为解；否则要继续S5的操作。

S5：以n作为新的m（即使27⇒m），以r作为新的n（即使6⇒n），求新的余数r(3)。

S6：判断上一个r是否等于零：若r＝0，则前一个n即为解；否则执行下面的操作。

S7：再以n作为新的m（6⇒m），r作为新的n（3⇒r），求新的r（r＝0）。

S8：判断上一个r是否等于零。这里已有r＝0，所以算法结束，n＝3即为60与33的最大公约数。

这种算法称为辗转相除。算法中使用的基本操作：整除求余和判断。它们都是进行算术运算时所允许的操作，只不过判断这一操作是在人脑中完成的，并不需要显式地写出来罢了。

从上面的算法中我们可以看出，S8、S6、S4与S2所完成的操作是相同的，S7、S5与S3所完成的操作也是相同的。或者说，S8、S6、S4是S2的重复，S7、S5是S3的重复。经过加工整理，可以得到下面的一个简单明了的算法2：

S1：置数。将两个数中的大数放到m中，小数放到n中。

S2：求余。求m/n的余数r。

S3：判断。若r＝0，则n就是所求最大公约数；若r≠0，执行下一步。

S4：置换。使新的m值作为n的值，使新的n值作为r的值，返回S2执行。

这个算法的S4，使用了循环，控制S2、S3、S4重复执行几次，使算法变得简单明了。稍作变换，我们还可以将它变换为下面的算法3：

S1：置数。将两个数中大数放到m中，小数放到n中。

S2：重复执行下面的序列，直到求得r＝0为止。

S2.1：求余。求m/n的余数r。

S2.2：置换。使m的值为n值，使n的值为r的值。

S3：输出n。

我们注意到，上面的算法对任何m与n都是适用的。要理解这个算法或要写出这个算法需要经过一定的抽象，即从具体问题出发，经过提炼，归纳，找出解决本类问题的普遍规

律。譬如在智力竞赛时，对于“1＋2＋…＋100”这样一个题目，有人按①作1＋2＝3，②作3＋3＝6，③作6＋4＝10，……的方法去作；而有人找到了另一种高效率的算法：100＋(1＋99)＋(2＋98)＋…＋(49＋51)＋50，共得50个100，1个50，即和为5050。一旦得到这样一种方法，当遇到题目为1＋2＋…＋1000时，他决不会重新冥思苦想地另寻算法，而是运用前面的方法，很快算出1000＋(1＋999)＋(2＋998)＋…＋(499＋501)＋500＝500500。

例 1.1.6　求正整数A和B的积，A和B为任意位数。

我们先看一下人工计算是怎样进行的。假如A＝3412，B＝513，乘法竖式为

```
  3412——A
 ×513——B
 -------
  10236
   3412   } 部分积
 17060
 -------
1750356——积
```

它们是这样进行的：以A为被乘数，分别以B的个位数、十位数、百位数作为乘数，求出它们的部分积。个位数与A相乘，得到第一行部分积，十位数与A相乘得到部分积（第二行）应左移一位，或者认为将该位数字乘以10再和A相乘；同样由乘百位数得到的第三行部分积应左移二位，或者认为将该数乘以100再和A相乘。把各部分积相加即得最后的积。

现在A和B的位数都是任意的，那么怎样用一般抽象的方法表示此算法呢？

将乘数B表示为$b_nb_{n-1}\cdots b_2b_1b_0$，$b_0$表示个位数字，$b_1$表示十位的数字，$b_2$表示百位的数字，其余类推。用C表示某一时刻的部分之和。用i表示乘数的位数，i＝0表示个位（因为$10^0=1$），i＝1表示十位（因为$10^1=10$），i＝2表示百位（因为$10^2=100$），……。

算法可以表示如下：

S1：使i＝0，C＝0（未进行乘数运算前，部分积之和为零）。

S2：当i≤n时，重复执行下面的操作：

S2.1：$C+A\times b_i\times 10^i\Rightarrow C$（乘$10^i$即左移i位，第一个部分积为$0+A\times b_0\times 10^0=A\times b_0$）；

S2.2：$i+1\Rightarrow i$（i增1）。

S3：输出C。

请读者按上述算法进行853×28654的运算。

例 1.1.7　求两个整数X(X≥0)和Y(Y＞0)的整数商和余数(规定只能用加法和减法运算)。

除法运算实际上是用加法和减法来实现的。其办法是，从X中一次又一次地减去Y，直到X＜Y为止，所减的次数即为商。例如5÷2的商为2，余数为1。可以将5先减2，余3，再减一次2，余1。每减一次，商加1。5能减两次2，所以商就得2。此时余数为1，不能再减2，结束。用Q代表商，R代表余数，算法可以表示为

S1：使Q＝0，R＝X（在开始减以前，商为零，所余的数就是X本身）。

S2：当R≥Y时，重复下面的操作：

S2.1：$R-Y\Rightarrow R$；

S2.2：$Q+1\Rightarrow Q$。

S3：输出R。

上面我们举了几个数学算法的例子。每一个算法都由加、减、乘、除判断、置数等基本操作，按顺序、判断分支、重复等结构组成。一般说来，任何复杂的数学问题都可以由上述这些最基本的操作按一定的结构组成的算法来求解。研究算法就是研究怎样把各种类型的问题的求解过程分解成上面的基本操作。

实际上，算法可以看作与演绎平行的另一数学体系。

由前面的讨论可知，算法是对解题过程的抽象和精确描述。一般来说，一个算法应当具备以下几个特征：

1．有穷性：一个算法应当包括有限个操作步骤，或者说它是由一个有穷的操作系列组成，而不应该是无限的。例如，创作一首小提琴曲谱可以称为小提琴算法。如果有人说他设计一首永远演不完的小提琴曲谱，人们将无法演奏完这个曲子，因而它不能称为算法。

不仅如此，有穷性还常常理解为实际上能够容忍的合理限度。例如，执行一个计算机算法需要让计算机运行10000年，便是不能容忍的。

2．确定性。算法中的每一步的含义都应该是清楚无误的，不能模棱两可，也就是说不应该存在“歧义性”（即可作两种以上不同的解释）。请读者分析下面这句话：张三对李四说他的孩子考上了大学。这句话就是“歧义性”的，可以理解为张三的孩子考上了大学，也可理解为李四的孩子考上了大学。因此在表示算法时要使用明确的文字或数字语言。

3．一个算法应该有零个或多个输入。如例1.1.5（求最大公约数）中，有两个输入量m和n。有的算法也可以没有输入。例如一盒音乐磁带，只要启动，就会放出音乐，而没有输入。

4．一个算法应该有一个或多个输出。算法的目的是求解，“解”就是输出。对无解的问题也应给出“无解”的信息。没有输出的算法是没有意义的。例1.1.5中的输出是最大公约数。

5．有效性。即我们在前面已经提到的一个算法必须遵循特定条件下的解题规则，组成它的每一个操作都应该是在特定的解题规则中允许使用的、可以执行的，并且最后能得出确定的结果。如在例1.1.5中给定的n为零时，执行第一步便会遇到一个在算术运算的条件下是无意义的运算（分母为0）。只要算法中有一个操作是不可执行的，整个算法就不具有有效性。

基于算法的特征，我们还可以给出算法的另一种定义：算法是一个过程，这个过程由一套清楚的规则所组成，这些规则指定了一个操作顺序，以便用有限的步骤提供特定类型问题的解答。这里需要指出的是，并非所有的过程都能称为“算法”。过程可以不受“有穷性”的约束，即它的操作可以是无穷尽的。例如求所有自然数之和，它要执行无穷尽的加法，那么这个工作便是一个“过程”，而不能称为算法。

对使用算法的人（而不是设计算法的人）来说，可以把一个算法看作一个“黑箱”，你给它输入某些确定的量，它就会输出确定的输出量。如求m和n最大公约数的算法的“外部特性”可以用图1.2表示。也就是说，从算法的外部（而不是研究算法的内部）看，它的作用是：给它输入m和n，它就能输出一个最大公约数。从此例也可以理解算法的输入和输出的概念。

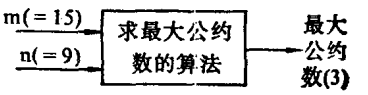

图 1.2

1.1.2 计算机——实现算法的有力工具

前面我们讨论了算法的基本概念，但是算法只是指出操作的内容和步骤，或者说只给出

了实现目标的过程描述，要把它付诸实践得到预期结果，还需要解决工具问题，即用什么工具去实现算法。正如用珠算算题，口诀是算法，算盘是工具一样。只有口诀没有算盘，就难以有效地算题。再如，作曲家写出一首名曲，用五线谱形式发表，这就是算法。但是五线谱不会发声，需要某种乐器，按照乐谱演奏出来，这就叫做“算法的实现”。最古老的算法工具是人工方式。用人工方式处理一些简单的算法，还能在可容忍的时间内完成，而对于复杂的算法，则往往人工难以在可容忍的时间范围内完成。例如，国外有人用手工计算π值，计算到小数点后的707位，花了十五年时间。如果要计算到小数点后的5000位，恐怕一辈子也完不成。

人类从远古时代便开始制造工具，以放大或延伸自身的能力。计算工具是一种智力工具。从棍石记事、屈指计算开始，人类先后制造出了算筹、算盘、计算尺、手摇计算机、电动计算机等计算工具，用以“放大”自身的智能，提高智力劳动的效率。1946年，世界上第一台电子计算机 ENIAC 诞生了。这是当时已经相当成熟与丰富的计算理论与电子技术相结合的硕果，也是当时正在迅猛发展的科学技术的迫切需要。ENIAC 是在美国陆军部主持下，由宾夕法尼亚大学电子系工程师J.P.Eckert和物理学家J.W.Mauchly 等人研制成功的。它的运算速度比当时已有的其他类型的计算机提高了一千倍，每秒钟可进行 5000 次加法、300 次乘法。四十年代末以来，电子计算机一直在突飞猛进地发展，已经经历了电子管、晶体管、集成电路、大规模集成电路等四代。运算速度已可达到每秒亿次以上。这样高的运算速度使其之前的任何计算工具及人的智力所望尘莫及，也使得我们能在允许的时间内完成过去无法实现的许多算法，如前述要把π的值计算到小数点后面5000 位等问题，都可以在短时间内实现。

然而，电子计算机的优势还并不单单在于其运算速度高上。它区别于其前那些计算工具的另一惊人成就是它能自动地执行操作。这一成果主要归功于一位美籍匈牙利数学家冯诺曼（Von Neumann）。

1944年夏天，正在普林斯顿工作的冯诺曼偶然地得到了正在研制 ENIAC 的消息，并引起了他的极大兴趣。他很快发现 ENIAC 的致命缺点，提出了以二进制和程序存储控制为基础的计算机体系结构思想。

二进制原理以二元逻辑为基础。它要求所有信息在机器内部都用0,1两个码的组合表示。表1.1是几个普通十进制数的0,1码表示。它遵循逢二进一的规律，也称二进制数。二进制原理确立了电子计算机经济而实用的物理结构。

表 1.1

十进制数	二进制数	十进制数	二进制数
0	0	5	101
1	1	6	110
2	10	7	111
3	11	8	1000
4	100	9	1001

程序存储控制理论使算法的自动执行成为可能。程序是以机器能理解的信息描述的算法。以前，程序被保存在机器外部，机器要由人按程序的规定步骤一步一步地操作，如用算

盘进行计算时，每一步运算都要由人按口诀内容拨动珠子进行。这就把人与机器捆在了一起，不仅把人束缚在繁冗的运算过程之中，而且难以发挥出机器运算速度高的这一突出优势。程序存储控制原理就是要让机器“记住”程序，并按程序规定的步骤控制自己的工作。这样便可以不需要人再去干预运算过程，解放了人，同时使机器的运算速度高的优势得到充分发挥。

现代计算机基本上都是按冯诺曼原理制造的，所以也称为冯诺曼型计算机。按照程序存储控制原理，现代计算机应由输入设备、输出设备、运算器、控制器、存储器等五大部件组成，如图1.3所示。当人用输入设备把程序和数据送到机器后，机器就把它们保存到存储器中。执行时，控制器从主存储器中依次取出程序指令，并逐条进行分析，根据指令的要求控制运算器对指定的数据进行相应的运算。控制器和运算器是计算机的心脏，由它进行控制与操作，它称为中央处理器（Centre Prosessing Unit），简称CPU。

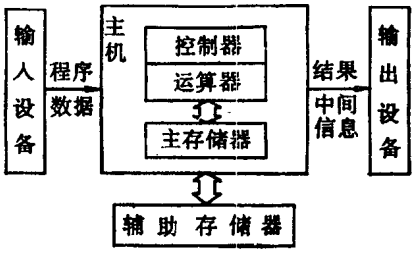

图 1.3

输出设备能将运算的中间信息（如出错、中间结果等）和最终结果输出到某个输出设备上。常用的输入设备是键盘，输出设备是显示器和打印机。

在计算机的全部设备中，存储器是一个很重要的设备。就像人没有记忆功能，就什么也做不成一样，计算机没有存储器，就无法存储程序和数据，就不能进行程序存储控制，自动工作也就无法谈起。

二进制电路是存储器的“细胞”。每个“细胞”可以存储一个二进制信息码，要么是“1”，要么是“0”，称为一“位”（b，即bit的缩写）。一个存储器可以存储亿万个二进制信息。为了管理方便，现代计算机把几个二进制信息组织在一起，称为一个“字节”（B，即byte的缩写）。一般一个字节由八位组成。存储器的容量一般以字节为单位，如长城GW0520DH的存储器可供用户使用的容量为640kB（$1k=2^{10}=1024$），而长城GW286B为1MB（$1M=2^{20}$，约为1兆），IBMPC为64kB，IBMPS/2-30为640kB，IBM PS/2-80为2～16MB。计算机中的一个逻辑信息，即一条指令或一个数据，称为一个计算机字（word）。一个字所包含的二进制位的个数称为字长。通常字长是字节的整数倍。如True BASIC规定每个数值数据占8个字节。

多数计算机是以字节为单位进行信息存储的。存储一个数据的存储空间，称为一个存储单元。一个存储单元可以包括一个字节或几个字节。为了能向指定的存储单元存入或取出指令或数据，需要对每一个存储单元编号，如同旅馆中的房间号一样。存储单元的编号，称为该单元的“地址”。按地址检索信息，是当代计算机存储器的一大特征。即向存储器中存入一个信息时，应指出把该数据存入哪个单元（指出地址），而从存储器中取一个数据时，是指取出哪个单元内的数据，而不是指取出哪个数值。如图1.4所示，假设在2001单元内已放入了数据“2”，在2002单元中已放入了数据“3”，如果想进行2+3的运算，应说“将2001单元中的内容与2002单元中的内容相加，送到2003单元中”，记作

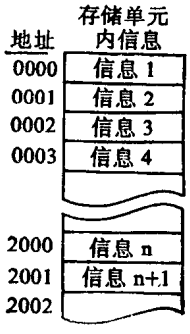

图 1.4

$$(2001)+(2002)\Rightarrow 2003$$

（2001）表示2001单元中的内容，而2003指单元地址而不是内容。请读者不要混淆单元地址