

C Y U Y A N C H E N G X U S H E J I J I A O C H E N G

C

语言程序设计 教程

潘广贞 主编

赵利辉 李玉蓉 副主编



国防工业出版社

National Defense Industry Press

1392741

C 语言程序设计教程

潘广贞 主编

赵利辉 李玉蓉 副主编

国防工业出版社

· 北京 ·

1475881

内 容 简 介

本书针对软件工程专业对程序设计的要求,强调以结构化程序设计思想为导向、以任务为驱动,书中例题都严格按照编码规范设计,并调试通过。主要内容包括:概述、C 语言数据类型、运算符和表达式、结构化程序设计(顺序、选择和循环结构程序设计)、数组、函数、指针、结构体与共用体、文件、位运算和预处理命令,还包括两个附录:C 语言运算符优先级和结合性以及 ASCII 代码与字符对照表。每章有本章的学习内容、基本要求及小结,并附有习题供读者练习。

本书是作者多年教学和软件开发实践经验的总结,具有内容丰富、叙述清晰、实用性强的特点。适合作高等院校计算机、软件工程等专业的程序设计基础课程教材,也可供对 C 语言及软件开发感兴趣的自学者使用。

图书在版编目(CIP)数据

C 语言程序设计教程/潘广贞主编. —北京:国防工业出版社,2010.8

ISBN 978-7-118-07057-6

I. ①C... II. ①潘... III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 173734 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京奥鑫印刷厂印刷

新华书店经售

※

开本 787×1092 1/16 印张 19 字数 464 千字

2010 年 8 月第 1 版第 1 次印刷 印数 1—4000 册 定价 37.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

前 言

在过去的很多年里,我们的学生一直在问这样一个问题:“老师,程序设计的思想和方法理论要应用到实践中去,怎么应用?除了那些枯燥的数字,除了求素数、分段函数……,我们对玩游戏其实更有兴趣,我们能用程序解决这些问题吗?”“没问题,可以!”但是对于同学们的很多切实的需求,我们要为他们推荐很多资料让他们去参考、学习,因为课堂上、教材里、实验中没有哪个环节能醍醐灌顶地向他们阐明这个问题。

为了让读者,尤其是初学者能更有兴趣地学习C程序设计语言,能让他们真正地学以致用,而不仅仅是为了学习,经过近两年半的努力,我们编写了《C语言程序设计教程》和《C语言程序设计实验指导》。在编写过程中,编者查阅并参考了大量的资料,针对计算机相关专业特别是软件工程专业的特点,设计了这两本书的结构及内容,书中每个例题都经过精心的选择,书中每个程序都进行了详尽的调试和测试,保证了程序的正确性。

但“万事开头难”,尤其是初学者,面对问题时在计算机面前往往束手无策,有时认为“这是一个不可能的任务”,但是基本的程序设计不需要那么多高等数学、线性代数等高深的数学知识和模型;将问题由陈述过渡到解决,只需要有勇气坐下来,用计算机能够理解的方式表述自己的思想。为此,本书主要面向计算机语言的初学者,为了增强读者的动手能力,并养成良好的编程习惯,与《C语言程序设计教程》配套的《C语言程序设计实验指导》将同步发行。实验指导从软件工程的角度出发,使学生在程序设计语言学习的第一时间就建立起使用规范的、可度量的软件工程方法和思想来解决实践中的问题,而不是在大学三四年级过于集中地学习软件工程的有关课程。从我们多年的教学经验看出,良好的程序设计思想、观念不是一朝一夕养成的,系统、规范和良好的习惯要从大学入学就开始培养和灌输。但是在学习的过程中困难是不可避免的,尤其对于初学者来说更是如此。但须记住,在学习的过程中一定不要被困难吓到,当一个“难题”被攻克的时候,成功的喜悦会抵消这个过程中所遭受的所有挫折和磨难。

“千里之行,始于足下”,和所有其他的技能一样,程序设计技术是一个循序渐进的过程,不可能一蹴而就,作为一个学习的起点,本书在内容安排上力求面向初学者,如在教程中的第1章详细讲述计算机硬件及其发展历史、C语言的发展及特点,以及算法、流程图等相关内容,给初学者的学习做充分的铺垫。为了帮助读者能够很好地掌握每章的内容,在每一章都设计了“学习内容”、“基本要求”和“小结”等,给读者以总体的把握和全面的总结。每章后有相应的习题,供读者对章节内容加强巩固,同时也是读者衡量自己对本章内容掌握情况的一个标杆。

虽然我们对本书进行了严格的审阅,但书中仍然会存在一些问题及不足,望广大读者不吝赐正。同时本书中参考引用了一些相关书籍和网站的内容,有些已在书后列出,在此对这些列出和未列出的内容所涉及的作者表示感谢,感谢您的思想、智慧对本书的成稿起到了关键作用。

同时,还要感谢对本书内容提出建议和意见的老师和同学。参与本书及配套实验指导编写的教师主要有付东来、康君、李玉蓉、王秀娟、薛海丽和赵利辉。各位编者付出了艰苦的劳动,在此表示深深的谢意。

编者

2010年7月

目 录

第1章 概述	1
1.1 计算机硬件及其发展	1
1.1.1 计算机发展简史	1
1.1.2 计算机工作原理	5
1.1.3 现代计算机硬件系统的构成	6
1.1.4 计算机软件系统的构成	7
1.2 程序设计语言	8
1.3 C语言简介	9
1.3.1 C语言的特点	10
1.3.2 C语言的发展和标准化	11
1.4 简单的C程序	11
1.4.1 C程序的特点	11
1.4.2 C语言程序的编译和运行	14
1.4.3 C语言程序的书写格式	16
1.5 解决实际问题的过程	17
1.5.1 解决问题的基本步骤	17
1.5.2 算法的概念	17
1.5.3 算法的特性	18
1.5.4 算法的表示	19
1.6 小结	21
习题	21
第2章 C语言数据类型、运算符和表达式	22
2.1 基本字符、关键字和标识符	23
2.2 常量	24
2.3 变量的定义和使用	26
2.3.1 变量的定义及变量类型	26
2.3.2 变量的属性和分类	27
2.3.3 变量的使用及赋初值	28
2.4 基本数据类型及其在计算机中的表示	29
2.4.1 数的进制	30

2.4.2	整型数据	35
2.4.3	整型变量	36
2.5	实型数据	40
2.5.1	实型常量的表示方法	40
2.5.2	实型变量	40
2.6	字符型数据	41
2.6.1	字符型常量	41
2.6.2	字符变量	42
2.6.3	字符变量在内存中的存储方式	42
2.6.4	字符数据和整型数据的关系	43
2.6.5	字符串常量	44
2.7	不同运算中各种数据类型间的相互转换	45
2.8	运算符和表达式	47
2.8.1	算术运算符和算术表达式	48
2.8.2	赋值运算符和赋值表达式	53
2.8.3	关系运算符和关系表达式	55
2.8.4	逻辑运算符和逻辑表达式	56
2.8.5	条件运算符和条件表达式	57
2.8.6	逗号运算符和逗号表达式	59
2.8.7	其他运算符及其运算	60
2.9	小结	60
习题		61
第3章	结构化程序设计	62
3.1	什么是结构化程序设计	62
3.2	程序流程图	63
3.2.1	传统流程图	63
3.2.2	N-S 结构化流程图	64
3.3	3 种基本结构及流程图表示	65
3.3.1	顺序结构	65
3.3.2	选择结构	65
3.3.3	循环结构	66
3.4	小结	66
习题		66
第4章	顺序结构程序设计	67
4.1	字符型数据的输入/输出函数	67
4.1.1	putchar 函数	67

4.1.2	getchar 函数	68
4.2	格式输入/输出函数	69
4.2.1	printf 函数	69
4.2.2	scanf 函数	75
4.3	程序举例	79
4.4	小结	81
	习题	81
第 5 章	选择结构程序设计	83
5.1	if 语句	83
5.1.1	if 语句的标准格式	83
5.1.2	if 语句的嵌套	85
5.2	switch 语句	89
5.3	程序举例	91
5.4	小结	96
	习题	96
第 6 章	循环结构程序设计	98
6.1	循环结构问题的提出	98
6.2	for 循环	98
6.2.1	for 循环语句	98
6.2.2	for 循环的简单举例	100
6.3	while 循环	101
6.3.1	while 循环语句	101
6.3.2	while 循环的简单举例	103
6.4	do-while 循环	103
6.4.1	do-while 循环语句	103
6.4.2	do-while 循环的简单举例	104
6.5	循环语句的嵌套	105
6.5.1	循环语句的嵌套形式	105
6.5.2	循环语句的嵌套举例	105
6.6	break 语句和 continue 语句	106
6.6.1	break 语句	106
6.6.2	continue 语句	108
6.7	程序举例	109
6.8	小结	115
	习题	115

第7章 数组	117
7.1 一维数组	117
7.1.1 一维数组的定义	117
7.1.2 一维数组的初始化	118
7.1.3 一维数组元素的引用	118
7.1.4 一维数组程序举例	120
7.2 二维数组	122
7.2.1 二维数组的定义	122
7.2.2 二维数组的初始化	123
7.2.3 二维数组元素的引用	124
7.2.4 二维数组程序举例	125
7.3 字符数组	126
7.3.1 字符数组的定义、初始化及引用	126
7.3.2 字符串处理函数	129
7.3.3 字符数组程序举例	131
7.4 小结	133
习题	133
第8章 函数	135
8.1 C语言程序的一般结构	135
8.2 函数的定义和返回值	136
8.2.1 函数的定义及声明	136
8.2.2 函数的返回值	140
8.3 函数间的数据传递	141
8.3.1 实参与形参	141
8.3.2 值传递与地址传递	142
8.3.3 简单变量作参数	146
8.3.4 数组作参数	148
8.4 函数的调用	152
8.4.1 函数调用的语法要求	152
8.4.2 函数的嵌套调用	152
8.4.3 函数的递归调用	153
8.5 变量的作用域及其存储类型	158
8.5.1 变量的作用域	158
8.5.2 变量的存储类型	161
8.5.3 变量分类总结	166
8.6 内部函数和外部函数	167

8.6.1	内部函数	167
8.6.2	外部函数	169
8.7	小结	171
习题		171
第9章	指针	173
9.1	指针的概念	173
9.1.1	变量的地址	173
9.1.2	变量的访问方式	174
9.1.3	指针和地址	174
9.2	指针变量	175
9.2.1	指针变量的定义	175
9.2.2	指针变量的初始化	175
9.2.3	指针的基本运算	176
9.2.4	指针程序举例	177
9.3	指针变量作函数参数	179
9.4	指向数组的指针	182
9.4.1	指向一维数组元素的指针	182
9.4.2	数组名或指针变量作函数参数	187
9.4.3	二维数组的指针	189
9.5	指向字符串的指针	195
9.5.1	指向字符串的指针变量	195
9.5.2	字符串指针作函数参数	198
9.6	指针数组	201
9.6.1	指针数组的概念	201
9.6.2	命令行参数	204
9.7	函数的指针和返回指针值的函数	206
9.7.1	函数的指针	206
9.7.2	指针型函数	209
9.8	指针的指针	212
9.9	小结	213
习题		214
第10章	结构体与共用体	215
10.1	结构体概述	215
10.1.1	结构体的概念	215
10.1.2	结构体类型的定义	215
10.2	结构体变量	217

10.2.1	结构体变量的定义	217
10.2.2	结构体变量的初始化	218
10.2.3	结构体变量的引用	219
10.3	结构体变量作为函数参数	220
10.4	结构体数组	222
10.4.1	结构体数组的定义	222
10.4.2	结构体数组的初始化	223
10.4.3	结构体数组元素的引用	224
10.5	指向结构体类型数据的指针	225
10.5.1	指向结构体变量的指针	225
10.5.2	指向结构体数组的指针	227
10.5.3	结构指针变量作函数参数	228
10.6	动态存储分配	229
10.7	链表	231
10.7.1	链表的概念	231
10.7.2	链表的操作	232
10.8	共用体	239
10.8.1	共用体类型定义	240
10.8.2	共用体变量的定义与引用	240
10.8.3	应用举例	241
10.9	枚举类型	244
10.9.1	枚举类型的定义和枚举变量的说明	244
10.9.2	枚举类型变量的赋值和使用	245
10.10	类型定义符 typedef	246
10.11	小结	247
	习题	248
第 11 章	文件	251
11.1	文件概述	251
11.1.1	C 文件的概念及分类	251
11.1.2	文件指针	252
11.2	文件的打开与关闭	253
11.2.1	fopen 函数	253
11.2.2	fclose 函数	254
11.3	文件的读写	255
11.3.1	字符读写函数 fgetc 和 fputc	255
11.3.2	字符串读写函数 fgets 和 fputs	257
11.3.3	数据块读写函数 fread 和 fwrite	259

11.3.4	磁盘文件读写函数 fscanf 和 fprintf	261
11.3.5	文件的定位	263
11.4	文件检测函数	266
11.5	C 语言库文件	266
11.6	小结	268
	习题	268
第 12 章	位运算	269
12.1	位运算符	269
12.1.1	按位与运算	269
12.1.2	按位或运算	269
12.1.3	按位异或运算	270
12.1.4	求反运算	270
12.1.5	左移运算	270
12.1.6	右移运算	270
12.2	位段	271
12.2.1	位段的定义和位段变量的说明	271
12.2.2	位段的使用	272
12.3	小结	273
	习题	273
第 13 章	预处理命令	275
13.1	概述	275
13.2	宏定义	275
13.2.1	不带参数的宏定义	275
13.2.2	带参数的宏定义	278
13.3	文件包含	281
13.4	条件编译	283
13.5	小结	284
	习题	285
附录 A	C 语言运算符优先级和结合性	286
附录 B	ASCII 代码与字符对照表	288
参考文献		290

第1章 概述

计算机能够自动、高速、精确地对各类数值或非数值信息进行存储、传送和处理,极大地推动了人类社会的发展与进步,对人类的生产和生活产生了重大的影响。目前,计算机已经融入到人们生活、工作和学习的各种场合,并且正在向更深的领域渗透。计算机之所以能够应用得如此广泛,除了20世纪以来计算机硬件技术的持续和高速发展之外,更离不开人们为了解决各种实际问题而编制的各种各样能够让计算机自动执行的程序(或指令序列)。今天,把程序连同其有关的说明资料统称为软件。可以毫不夸张地说软件是计算机的灵魂,软件使计算机使用起来如此丰富多彩,是软件使计算机功能如此强大,如果没有软件,计算机就如同没有MP3的播放器和没有闭路信号的电视机一样,与废铁没什么差别。使用不同的计算机软件,计算机可以完成许许多多不同的工作。它使计算机具有非凡的灵活性和通用性。也正是这一原因,决定了计算机的任何动作都离不开由人安排的指令。人们需要为他们的不同需求而编制不同的计算机程序。今天人们使用的各类计算机、手机、MP3播放器、PDA、导航设备,甚至电梯、汽车等无不需要为它们量身定制的程序,程序的功能之强大由此可见一斑。那么,在开始学习程序设计之前,读者应首先了解什么是程序、什么是编写程序所需的算法以及如何使用流程图对算法进行描述等基础性问题。本章旨在使读者对程序设计和C程序设计语言有一个初步的认识,为以后的学习打下基础。

【学习内容】C语言是目前最流行的程序设计语言和教学语言之一,它适合用作系统描述语言,可以用来开发系统软件和应用软件。本章首先介绍计算机的发展简史,在此基础上引出C语言发展的历程,系统地介绍C语言的特点、发展历程和标准化过程。然后通过例子给出C语言从源文件的编辑到预处理、编译、连接和生成可执行文件的详细过程。同时,介绍C语言源文件编辑中的格式、注释问题并着重指出如何利用C语言解决实际问题的方法。

【基本要求】理解硬件和软件的区别;认识问题求解是计算机科学的一个重要部分;理解常用术语的含义;认识软件维护的重要性和使用良好的软件工程实践方法的重要性;理解C语言源程序的编译过程;掌握C语言解决问题的思路和常见问题的处理方法。

1940年以前,计算机还不存在;而今天,计算机却已渗透到人们学习、生活和工作的各个角落。尤其随着现代网络技术的发展,计算机技术的发展和运用更是日新月异,层出不穷。因此,学习计算机对于人们的工作和生活有着重大的现实意义。

1.1 计算机硬件及其发展

1.1.1 计算机发展简史

每秒钟1206万亿次的峰值运算速度,随着2009年10月29日我国第一台千万亿次超级计算机“天河”一号(图1-1)的诞生,我国成为继美国之后世界上第二个能够自主研制千万亿次超级计算机的国家。这一事实表明,即使在多媒体技术、通信技术和信息技术大

行其道的今天，“计算”作为计算机最重要的功能之一仍然是衡量一个国家计算机技术的发展水平的标志之一。事实上，正是由于“计算”的推动才导致计算机的发明和应用。众所周知，“计算”并不是在计算机产生之后才有的，而是古已有之。

在中国，典型的“计算”代表是“算盘”（图 1-2）的发明和使用。算盘包括算具（硬件）、算法（软件、口诀、歌诀）两个方面。关于算盘的来历，最早可以追溯到公元前 600 年，据说我国当时就有了“算板”。古人把 10 个算珠串成一组，一组组排列好，放入框内，然后迅速拨动算珠进行计算。东汉末年，徐岳在《数术记遗》中记载，他的老师刘洪访问隐士天目先生时，天目先生解释了 14 种计算方法，其中一种就是珠算，采用的计算工具很接近现代的算盘。这种算盘每位有 5 颗可动的算珠，上面 1 颗相当于 5，下面 4 颗每颗当作 1。随着算盘的使用，人们总结出许多计算口诀，使计算的速度更快了。这种用算盘计算的方法，叫珠算。到了明代，珠算不但能进行加减乘除的运算，还能计算土地面积和各种形状东西的大小。由于算盘制作简单，价格便宜，珠算口诀便于记忆，运算又简便，所以，在中国被普遍使用，并且陆续流传到了日本、朝鲜、美国和东南亚等国家和地区。

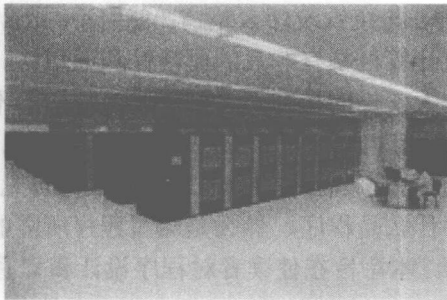


图 1-1 “天河”一号千万亿次超级计算机系统

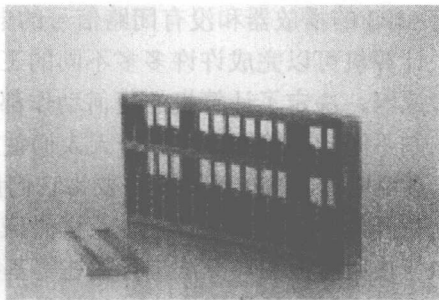


图 1-2 算盘

得益于算盘（算筹）的发明，我国的计算工具也曾经独领风骚。例如，祖冲之在南北朝时用算筹得到的圆周率精确到小数点后 7 位，促进了当时计算精度的提高，为以后各种工程项目的实现，新工具的发明、改进打好了基础。但由于各种各样的原因，发展的后劲不足，理论创新，技术创新不能一以贯之，缺乏积累，最终全面落后于西方科技。但是人类探索机器的算术能力从未因一个民族的停滞而停止，在欧洲，从一开始的计算尺，到法国人 Pascal 和德国人 Leibniz 设计的机械加法器、乘法器，再到按照连接顺序的打孔卡控制编制样式的织布机，再到 Babbages 设计的差分机，直到 IBM601、ABC、ENIAC、EDVAC，此时与现代计算机的设计思想相差无几，最终带来了信息技术革命。可以说现代社会的绝大部分便利都受惠于此。下面介绍这一时期以欧洲为代表的西方世界在“计算”领域的发展史，正是因为他们的孜孜追求所形成的理论与技术基础为计算机的成功发明起到了极大的作用。

1614 年，苏格兰人 John Napier 发表了一篇论文，其中提到他发明了一种可以进行四则运算和方根运算的精巧装置。

1623 年，德国科学家 Whihelm Schickard 发明了第一台机械计算器，这种计算器可以自动执行简单的算术运算。

1625 年，William Oughtred 发明计算尺。

1630 年，英国数学家奥特雷德使用当时流行的对数刻度尺做乘法运算，突然萌生了一个念头：若采用两根相互滑动的对数刻度尺，不就省得用两脚规度量长度吗？他的这个设想导

致了“机械化”计算尺的诞生。

1642 年, 年仅 19 岁的法国科学家帕斯卡 (Blaise Pascal) (图 1-3) 发明了机械加法机 (图 1-4)。帕斯卡加法机是一种系列齿轮组成的装置, 外壳用黄铜材料制作, 是一个长 20 英寸、宽 4 英寸、高 3 英寸的长方盒子, 面板上有一列显示数字的小窗口, 旋紧发条后才能转动, 用专用的铁笔来拨动转轮以输入数字。这种机器开始只能够做 6 位加法和减法。然而, 即使只做加法, 也有个“逢十进一”的进位问题。聪明的帕斯卡采用了一种小爪子式的棘轮装置。当定位齿轮朝 9 转动时, 棘爪便逐渐升高; 一旦齿轮转到 0, 棘爪就“咔嚓”一声跌落下来, 推动十位数的齿轮前进一档。



图 1-3 法国科学家 Blaise Pascal

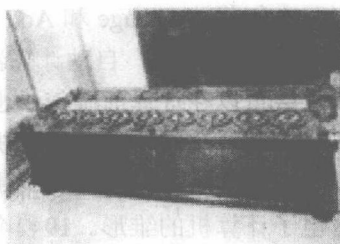


图 1-4 Blaise Pascal 发明的机械加法机

1673 年, 德国数学家 Gottfried Leibniz 发明了可以进行加、减、乘、除运算的复杂装置。所有上述发明都是纯机械的, 没有使用任何动力源。操作者将金属轮转到特定位置来输入数字, 而转动这些金属轮又可带动机器的其他部分, 从而显示不同的结果。

18 世纪, 源于英国的工业革命推动人们开始考虑使用蒸汽机驱动更加复杂的计算机器, 英国数学家、可编程计算机的发明者、计算机的先驱 Charles Babbage (图 1-5) 在这一时期设计了两种不同的计算器: 差分机 (图 1-6) 和分析机 (图 1-7), 这成为当时计算器发展进步的代表。但遗憾的是他没有能完成这些项目。他所设计的用于产生数学函数表的差分机直到 30 年后的 1854 年才由一个瑞典发明家实现; 同样抱憾的是, 直到 1871 年他逝世分析机的发明也未能最终实现。尽管如此, Babbage 的发明仍带有许多现代计算机的基本特点。比如, 他的分析机可根据设计好的程序实现不同的功能。在这个设计中, 分析机的操作是由一张卡片上的一组小孔来控制的。机器可以读出这组小孔的排列模式。通过改变小孔的排列模式, 改变机器的行为, 执行不同的运算。在这一时期, 令人难以忘记的是 Augusta Ada Byron。



图 1-5 Charles Babbage

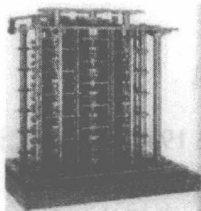


图 1-6 差分机

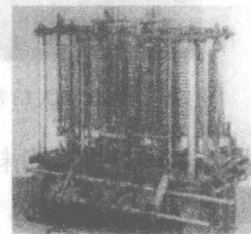


图 1-7 分析机

Augusta Ada Byron 是当时著名的才女, 也是著名诗人拜伦的女儿。Ada 不但能诗善画, 还是当时著名数学家 Augustus De Morgan 的学生。她因为某个机缘, 参观了 Babbage 的工作室。虽然 Babbage 的工作室内零件散落满地, 但屋子中央座落着的那台怪物般的半成品, 却

让 Ada 认为这是一个“伟大而美丽的发明”。Ada 参加了 Babbage 的一部分工作，尤其是程序设计方面。今天，人们认为最早的计算器应该归功于 Babbage，而最早的程序设计应该归功于 Ada。为此，20 世纪 80 年代由美国国防部委托设计的一套程序语言，就以 Ada 命名。

Charles Babbage 在分析机和差分机上的设计思想在很大程度上影响了 19 世纪的计算发展史。例如，1805 年，Jacquard 制造了可编程序型自动织布机。它使用不同式样的厚纸卡来控制纺锤和飞梭的交互作用，使其编出不同花式的布料。这个发明引发了纺织工业的一次革命。根据记录，到了 1812 年，这种机器一共卖出 1.1 万台。1890 年，Herman Hollerith 采用 Babbage 的设计，使用穿孔卡片为美国人口普查数据自动生成报表（图 1-8），为推广这一技术，他成立了一家公司，即后来著名的 IBM(国际商业机器)公司。

在 18 世纪 40 年代，Babbage 和 Ada 已经认识到：计算程序的核心技术在于重复。Babbage 和 Ada 曾经明确地写了下来：自动计算器的真正重要之处，在于它可以重复执行一套给定的程序。其重复次数可以在计算前确定，也可以依据计算结果而临时决定。

20 世纪 40 年代，电子学的发展使机械计算器的出现成为现实，也使 Babbage 的可编程计算机由梦想变为现实。1939 年衣阿华州立大学的 John Atanastoff 和他的学生 Clifford Barry 组装了第一台电子计算机的雏形。1942 年 5 月，他们又组装了一个完整的包含 300 个电子管的计算机，后来计算机改进的工作由于第二次世界大战而中断。

1946 年 2 月 15 日，世界上第一台电子数字式计算机在美国宾夕法尼亚大学摩尔学院的 J. Presper Eckert 和 John W. Mauchly 的指导下研制成功，它的名称叫 ENIAC（图 1-9），是电子数值积分式计算机（The Electronic Numerical Intergrator and Computer）的缩写。它使用了 17468 个真空电子管，功率 174kW，占地 170m²，重达 30t，每秒钟可进行 5000 次加法运算。虽然它还比不上今天最普通的一台微型计算机，但在当时它已是运算速度上的绝对冠军，并且其运算的精确度和准确度也是史无前例的。以圆周率 π 的计算为例，中国的古代科学家祖冲之利用算筹，耗费 15 年心血，才把圆周率计算到小数点后 7 位数。1000 多年后，英国人香克斯以毕生精力计算圆周率，才计算到小数点后 707 位。而使用 ENIAC 进行计算，仅用了 40s 就达到了这个记录，还发现香克斯的计算中，第 528 位是错误的。

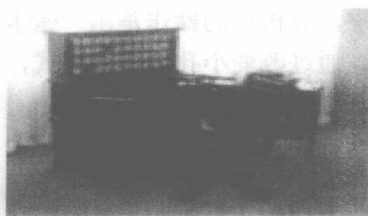


图 1-8 1890 年人口普查使用的制表机

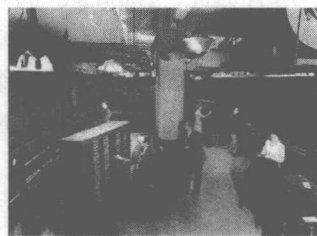


图 1-9 ENIAC

其实，在 ENIAC 尚未投入运行前的 1944 年，现代计算机之父，美国普林斯顿大学的美籍匈牙利科学家约翰·冯·诺依曼（John Von Neumann）（图 1-10）由于偶然的机会加入了 ENIAC 计算机研制小组，并且在 1945 年带领他的科研团队，发表了一个全新的“存储程序通用电子计算机方案”——EDVAC（Electronic Discrete Variable Automatic Computer）。冯·诺依曼以“关于 EDVAC 的报告草案”为题，起草了长达 101 页的总结报告（即计算机史上著名的“101 页报告”）。报告广泛而具体地介绍了制造电子计算机和程序设计的新思想。这份报告是计算机发展史上一个划时代的文献，它向世界宣告：电子计算机的时代开始了。至此

计算机的发展进入正轨，此后，“冯·诺依曼”计算机层出不穷，首先是 IBM 的 701、后来的 PC (Personal Computer) 兼容机、Apple 公司的 Macintosh 系列计算机等。今天，计算机已经走进千家万户，成为人们日常工作、学习和生活的重要组成部分。



图 1-10 冯·诺依曼

1.1.2 计算机工作原理

在冯·诺依曼的 101 页报告中，提出了 EDVAC 计算机的两大核心设计思想：二进制和程序内存。

“二进制”思想是他根据电子元件双稳工作的特点，建议在电子计算机中采用二进制，并预言采用二进制可以大大简化机器的逻辑线路。实践证明了诺依曼预言的正确性。如今，逻辑代数的应用已成为设计电子计算机的重要手段，在 EDVAC 中采用的主要逻辑线路也一直沿用着，只是对实现逻辑线路的工程方法和逻辑电路的分析方法作了改进。

“程序内存”是冯·诺依曼的另一杰作。通过对 ENIAC 的考察，冯·诺依曼敏锐地抓住了它的最大弱点——没有真正的存储器。ENIAC 只有 20 个暂存器，它的程序是外插型的，指令存储在计算机的其他电路中。这样，解题之前，必需先设计所需的全部指令，通过手工把相应的电路连通。这种准备工作要花几小时甚至几天时间，而计算本身只需几分钟。计算的高速与程序的手工操作存在着很大的矛盾。针对这个问题，诺依曼提出了程序内存的思想：把运算程序存在机器的存储器中，程序设计员只需要在存储器中寻找运算指令，机器就会自行计算，这样，就不必每个问题都重新编程，从而大大加快了运算进程。这一思想标志着自动运算的实现，标志着电子计算机的成熟，已成为电子计算机设计的基本原则。

冯·诺依曼在他的 101 页报告中还提出 EDVAC 计算机将由 5 个部分组成，包括：运算器、逻辑控制装置、存储器、输入和输出设备，并描述了这 5 部分的职能和相互关系，即数据通过输入设备或外存储器输入内存，内存中的数据通过运算器进行运算，运算的结果再返回到计算机的内存中，需要输出的数据通过输出设备输出或存储到外存储器中，而这一切过程均在控制器的控制下完成。这一理论被尊称为“冯·诺依曼”计算机原理（图 1-11），该原理奠定了现代计算机的理论基础和架构体系，而他本人也被誉为“现代计算机之父”，直到今天这一理论仍然沿用在几乎所有的微型计算机上。因此，了解“冯·诺依曼”计算机原理将有助于广大读者更深入地理解计算机的工作原理和它的软、硬件架构体系，对于大家学习程序设计也将大有裨益。

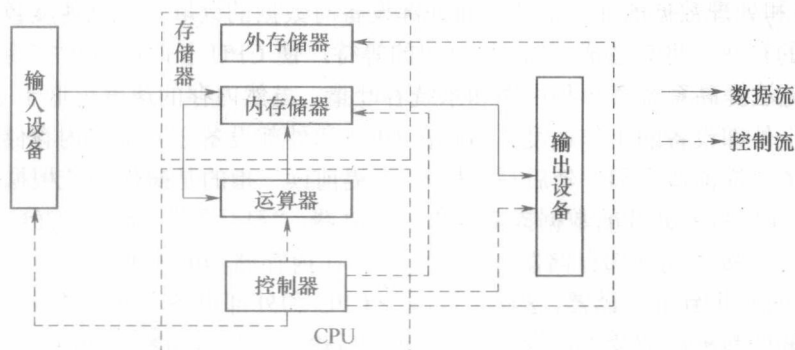


图 1-11 冯·诺依曼计算机工作原理图