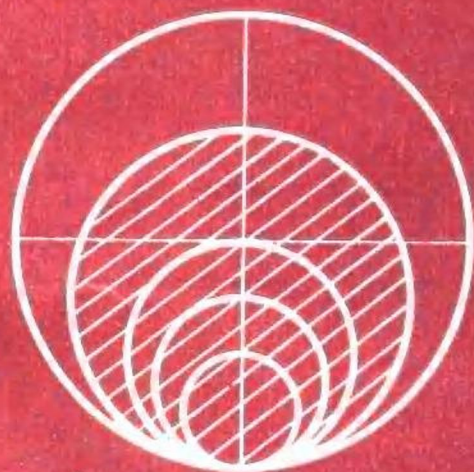


BASIC

绘图三日通

——让微机在办公室自动化中
发挥更有效的作用



测绘出版社

内 容 简 介

本书结合办公室自动化中的实际问题。通过简短的 BASIC 程序, 绘制制柱状图、直方图、带形图、环形图、折线图、雷达航线图等常见的实用图形。它从简单的 BASIC 程序设计开始, 逐步深入到现代科学管理中一些基本方法的应用。内容深入浅出, 通俗易懂。为适应国内的机型, 译者对书中的程序进行了二次开发, 可供读者直接应用。

本书可供从事微机应用的技术人员、企、事业单位的管理人员以及有关大、中专院校的师生参考。

BASIC 绘图三日通

——让微机在办公室自动化中发挥更有效的作用

(日) 工学博士 间野浩太郎 审修

(日) 竹内和夫 著

张树增 刘德周 编译

周贤鸿 校

*

测绘出版社出版

测绘出版社印刷厂印刷

新华书店北京发行所发行·各地新华书店经售

*

开本 787×1092 1/32·印张 6.5·字数 167 千字

1986 年 9 月第一版·1986 年 9 月第一次印刷

印数 0.001—8,000 册·定价 1.40 元

统一书号: 15039·新 551

译者的话

本书译自日文《BASIC 三日通——让微机在办公室自动化中发挥更有效的作用》，オーム社，1984年1月第一版。通过实践，我们认为间野浩太郎教授说这是一本难得的好书并非夸张之词。

这本书主要好在以下两个方面：一是立意图新，在让微机把所需数据制成一个表进行输出已成为一种传统方式的今天，本书不失时机地另辟溪径——普及与推广图形输出。由于把有关数据绘制成不同的图形，从而把问题反映得更加直观、更加清晰，而使得微机在办公室自动化中，能发挥出更加有效的作用；二是雅俗共赏。本书通过微机绘图这一手段，广揽办公室自动化和质量管理工作中的实际问题，从指导入门起，逐步引伸到现代科学管理的有关理论与方法。由于和大家所熟悉的日常实际紧紧相关，所以，既实用又易懂，做到了广而不散，深而不难。

正因为本书有上述两大特点，所以，无论对懂与不懂微机应用的人来讲，都能从这本书中获得不少的新知。我们深信，当大家读完这本书以后，一定会感到它的确是一本难得的好书。

由于本书的实用性很高，考虑到长城 0520、IBM-PC/XT 等机型在我国的微机领域中占有很大的比重，为减少许许多多人的重复劳动，迅速提高微机在办公室自动化等方面的应用水平，我们在长城 0520 机器上，在中文操作系统 CCDOS 的支持下，汉字屏幕为 40 字×25 行，使用 GW-BASIC 语言，对本书程序实例都进行了二次开发。本着尽量少改动的原则，把必须修改的内容编入书内。

承霍俊杰工程师在百忙中协助校阅译稿，并得到了王丽君等不少同志的帮助，特致以衷心的感谢。由于水平、学识有限，在编译方面难免存在着很多的不足，恳请读者指正。

刘德周 张树增 1985 年 10 月

前 言

微机作为实现办公室自动化——OA 的基础，已经在许许多多的企、事业单位得到广泛地应用。由于微机应用工作开展的好坏，与企业的盛衰紧密相联，所以，现在要求每一个企业人员了解(或掌握)微机在企业中的应用，是不过分的。

由于这本书的例题及其程序的设计思路都和大家所熟悉的实际问题结合在一起，而且又加了详尽的说明，所以，无论是对想学习 BASIC 语言的人来讲，还是对想把身边的微机更有效地运用到办公室自动化(OA)或质量管理(QC)等方面的人来讲，都可把它作为一本入门实践书。

大家都知道，微机在办公室自动化中的应用所涉及的范围极其广泛。本书的特点是：通过简短的程序绘制出常见的实用图形这一手段，来归纳有关知识。由于各个例题都是从一看就能明白的地方入手，所以身边有微机的话，一天运行几个程序例题是不难做到的。通过这样的实践便能自然而然地理解、掌握 BASIC 语言的功能和程序的编制方法。进而通过与各个程序例题紧密相关的“有关知识”和“应用问题”来扩大视野，以提高自己独立处理微机在办公室自动化应用方面的能力。

本书所使用的微机是 NEC 公司的 PC 8801 系统。由于各个程序都是用 BASIC 语言的基本命令编制的，所以，对于其它机型来说，应该完全适用。

在本书的编著和出版过程中，得到了审修人青山学院大学间野浩太郎教授以及欧木(オーム)出版社的很大帮助，借本书出版之际，特致以深深的谢意。

竹内和夫

1983 年 11 月

原 序

大家都知道，微机的应用领域不但十分之广，而且能在各个方面发挥很大的作用。可是对于大多数人来讲，尽管想用它，但是由于不知道怎样用它或怎样更好地用它，而使它发挥不出应有的作用。于是在报纸或杂志上经常出现诸如此类的消息：“为跟上时代的潮流，买回了微机，但是由于不知道如何使用而又发愁，最后不得不低价转手”。

出现这种情况的原因主要有两个：一是让微机处理的问题超过了原来的设想。由于程序太大或出现新的要求，必须进行硬件的扩充。例如需要增加字处理器和能打印日文的打印机等等。二是懂得编制程序的人太少了。别说解决能起很大作用的实际问题，就连非常短小的程序，很多人也编不了。

从微机常常当做游戏机可以清楚地看出，微机的价格不但十分便宜，而且它具有相当出色的绘图功能。这就是说，我们可以用微机画图形、画图表。

不管是让别人理解自己的意图，还是分析、总结自己的工作，把有关数据绘制成图表或图形，都是非常必要的。然而这对很多人来讲，并不是件轻而易举的事。如果有一个绘制某种图形的现成程序就好了，只要给定数据马上就能得到相应的图形。但是，大多数人认为买这种软件划不来，因为一来要花钱，二来也不知道它是怎样画出来的。

本书将给大家解决这一问题创造条件，带来方便。因为本书结合常见的实际问题，不但绘制出其相应的图形、列出绘图所用的程序清单，而且还进行了通俗易懂的说明。所以，对于没有什么编程知识的人来讲，无论阅读那一个章节，都能理解它的内容，而对于希望学会或提高 BASIC 程序设计能力的人来讲，则请您

从头开始阅读全部内容。

由于这本书同大量的日常实际问题相结合，所以它不仅适用于社会各界人士，而且对于经营、管理等专业的学生来说，也是一本极好的参考书。在此，特向各位推荐这本难得的好书。

间野浩太郎

1983年11月

目 录

一	随机数的采集 (1)	
	—— 随机数的产生和循环操作	(1)
二	随机数的采集 (2)	
	—— 数据的贮存和比较	(9)
三	选择工作时间	
	—— 数据的分类 (排序)	(14)
四	点图	
	—— 图形符号的使用	(20)
五	立柱状统计图 (1)	
	—— 显示器的屏幕坐标和坐标变换	(28)
六	立柱状统计图 (2)	
	—— 在图形上写入刻度	(38)
七	横向柱状统计图	
	—— 长方形的横置	(46)
八	带形统计图	
	—— 按比例分割长方形	(50)
九	面积统计图	
	—— 画圆	(54)
十	圆形统计图	
	—— 分割圆为扇形	(60)
十一	半环形统计图	
	—— 分割两个同心半圆	(65)
十二	折线统计图 (1)	
	—— 把相邻两点连接成直线	(73)
十三	雷达航线统计图	
	—— 把折线画在同心圆上	(79)

十四	折线统计图 (2)	
	—— 把三个项目画在同一图形上	(86)
十五	编制频数分布表	
	—— 按数量等级整理数据	(93)
十六	直方图	
	—— 求平均值和标准偏差	(98)
十七	频数多角形	
	—— 用折线图表示频数分布	(104)
十八	销售队列图	
	—— 掌握重点项目	(108)
十九	库存管理 (1)	
	—— 求安全库存量和订货点	(115)
二十	库存管理 (2)	
	—— 求最佳订货量	(119)
二十一	散布图	
	—— 观察两组数据的相互关系	(127)
二十二	回归分析	
	—— 用直线近似地表示两组数据的相关情况	(133)
二十三	趋向分析	
	—— 用回归方程式预测未来的需求量	(137)
二十四	需要预测的模拟	
	—— 根据概率分布建立需求量的模型	(142)
二十五	工程能力的调查	
	—— 判断工程能力的有无	(149)
二十六	$\bar{X}$ (平均值) 管理图	
	—— 分析加工过程的管理状态	(158)
二十七	线性规划法	
	—— 求效益最大时的生产量	(167)
	应用问题的解答	(177)
	索引	(199)

一 随机数的采集(1)

——随机数的产生和循环操作

(一) 实例的处理

现在有这样一个问题：从一百个产品中，任意地选取十个进行抽样检查。令产品编号由 0 到 99。

用计算机处理这个问题的 BASIC 程序及其执行结果如下：

```
10  FOR i = 1  TO  10
20  a = INT (RND * 100)
30  PRINT  a;
40  NEXT  i
50  END
run
95  42  80  2  86  88  43  19  50  80
Ok
```

(二) 程序的要点

为了丝毫不加杂人为意志地从待检产品中抽样，利用计算机取随机数是一个最好的方法。在这个问题中，由于产品编号是 0 ~ 99，所以应产生十个 0 ~ 99 之间的两位整数的随机数，做为被抽样产品的编号。于是，如何“产生一个两位整数的随机数”和“反复地进行 10 次这种操作，即循环 10 次。”便成为该程序所要解决的两个关键问题。

(三) 程序的说明

10 行	本行和 40 行之间构成一个循环操作,本行是循环的开始。 <i>i</i> 为循环的控制变量,它控制计算机把 10~40 行之间的程序行反复地执行 10 次。它的值从 1 开始,每从 40 行返回一次就加 1,一直到 10,即 <i>i</i> 的初值为 1,步长为 1,终值为 10。
20 行	生成两位整数的随机数,并将其值赋给变量 <i>a</i> 。
30 行	输出变量 <i>a</i> 的当前值。
40 行	本行为一次循环的结束处。如果 <i>i</i> 的值不大于 10,就返回到第 10 行进行新的一次循环。如果 <i>i</i> 的值大于 10,就退出循环,而顺序地执行下一个程序行。
50 行	程序结束。

1. FOR~NEXT 语句

为了反复地执行某一相同的操作,应使用 **FOR~NEXT** 语句。

这个语句将反复地执行用 **FOR** 语句和 **NEXT** 语句所围住的各个程序行(这就是所谓的 **FOR 循环**)。在 **FOR** 语句中规定了循环的次数。**FOR** 和 **NEXT** 语句的书写形式如下:

FOR 变量名 = 初值 TO 终值 STEP 步长
)
NEXT 变量名

当步长为 1 时,可以省略 **FOR** 语句中的 **STEP** 及其后面的短语,如本程序中行号为 10 的那个程序行。

下面的例子中循环的次数都是 5。

例 1.1

```
FOR a = 0 TO 4
)
NEXT a
```

例 1.2

```
FOR x1=0.1 TO 0.5 STEP 0.1
  {
NEXT x1
```

例 1.3

```
FOR n=-4 TO 0
  {
NEXT n
```

例 1.4

```
FOR k=50 TO 10 STEP -10
  {
NEXT k
```

2. PRINT 语句

该语句的功能是把它后面所指定的变量名或表达式的值输出到屏幕（显示器）上，其书写形式如下：

PRINT 输出信息

输出信息的写法有各种各样的形式，如下面的例题。

例 1.5

输出信息可以是表达式，所以我们把上述的程序的 第 20 行和第 30 行合并起来，写成下述的形式也是可以的。

```
20 PRINT INT (RND*100) ;
```

下面我们再看看 **PRINT a** 和 **PRINT a;** 的区别。

例 1.6

```
10 FOR a=1 TO 10
20 PRINT a
30 NEXT a
```

该程序执行后，将在屏幕的**竖直方向**上顺次输出 a 的值 1，2，3，……，10。共占 10 行。

例 1.7

```
10 FOR a=1 TO 10
```

```
20 PRINT a;
```

```
30 NEXT a
```

该程序执行后，将在屏幕的**水平方向**上并排输出一行 a 的值
1, 2, 3, ……，10。

3. END 语句

该语句的功能是宣布程序结束。**当把它放置在程序的最后一行时，可以省略。**因此，在以后的程序中，除非必要，都省略了 END 语句。

4. INT 函数

其书写格式为：

INT (表达式)

该语句的功能是对括号内的表达式的值，取小于或等于该值的最大整数。

5. RND 函数

该语句的功能是产生一个大于 0 而小于 1 的随机数，按如下形式书写：

RND

在上述程序中，编号为 20 的程序行的操作过程是：先由 RND 函数取得一个小于 1 的随机数，而后将其扩大 100 倍 ($RND * 100$ ，*** 是乘法运算符**，在此读作乘号)，再由 INT 函数舍去小数部分，于是就得到了一个大于等于 0 而小于等于 99 的随机整数，并将这个数存放于变量 a 中。

6. 命令语句和函数

FOR~NEXT 和 PRINT, END 等语句是指示计算机完成某一指定操作的命令，叫做**命令语句**。把这些命令语句按照某一特定工作的处理顺序书写出来，就是**程序 (PROGRAM)**。如图 1.1。

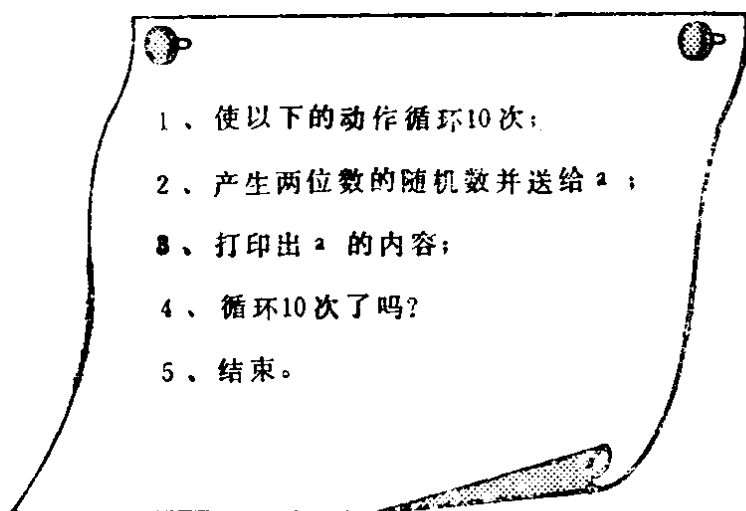


图 1.1 程 序

而对于 INT, RND……来讲, 由于它本身就是能独立完成一项工作的程序, 所以我们把它叫做**函数**。例如, RND 函数就是让计算机产生一个大于 0 而小于 1 的随机数的程序。为了提高程序编制者的工作效率, 计算机厂家都把一些常用的处理程序, 制作成标准函数, 如: **SQR**——是求平方根的函数, **SIN** 和 **COS**——用来求正弦和余弦的三角函数, **EXP**——指数函数等等, 以便供程序编制者直接引用。

7. 变量名

任何一个程序都要对数据进行各种处理, 以产生出人们所需要的数据, 因此需要把正在处理的数据和加工出来的数据暂时存放在计算机的**存贮器** (**记忆装置**, 即 **MEMORY**, 亦即存放程序和数据的场所) 里。为了正确区分不同数据的存放位置, 需要给不同的数据取不同的名字, 这个名子就叫做**变量名**。也就是说, **所谓变量名**, 就是象图 1.2 所示的那样, 实际上是**给存放不同数据的各个存贮场所**临时起的一个名字。上述程序中的 a 和 i 就是变量名。它们分别是存放随机数和循环控制变量的存贮场所的名字。

例如, 程序的第 20 行

$$a = \text{INT} (\text{RND} * 100)$$

就是这样的语句 (**STATEMENT**), 它把由右式得到的值存放在

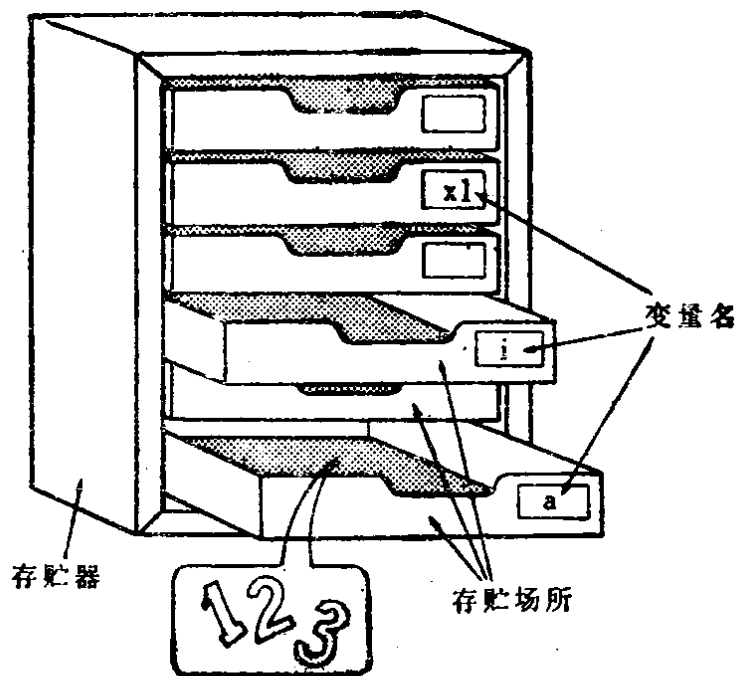


图 1.2 存储器与变量名

叫做 **a** 的存储单元里（**运算符“=”，不是“相等”的意思，而是把右边的内容“移给”左边，“存放”在左边的意思。**）。又如程序的第 30 行，

```
PRINT a;
```

它将把叫做 **a** 的存储单元存放的数据输出到屏幕上。

原则上可以使用任意的英文字母和数字作变量名，如 **a**，**i**，**x1**，**OHM**，**PC8800** 等等。在本书中，为了使程序清楚易懂，**命令语句和函数用大写字母表示；变量名用小写字母表示。**然而计算机并不区分大写字母和小写字母，用哪一种字体都行。

作为字符串使用的大、小写字母，输入计算机后，在屏幕上的显示和打印机输出均照原样，而作为变量使用的大、小写英文字母，输入计算机后，屏幕上可照原样显示，但是，打印机输出的确均为大写字母。

8. 变量名和地址

如图 1.3 所示，计算机存储器的各个存储单元已被赋与从 0 开始一连串编号，这些编号叫做**内存地址(ADDRESS)**。计算机对存储器进行数据的存、取操作时，都根据指定的地址来进行。

因此，所谓**变量名**，实际上就是为了代替内存地址而起的名字。用户使用变量名不必知道它具体使用的是哪个内存地址，只须记住变量名 **a** 是存放变量 **a** 的存贮地址，变量名 **x1** 是存放变量 **x1** 的存贮地址就行了。

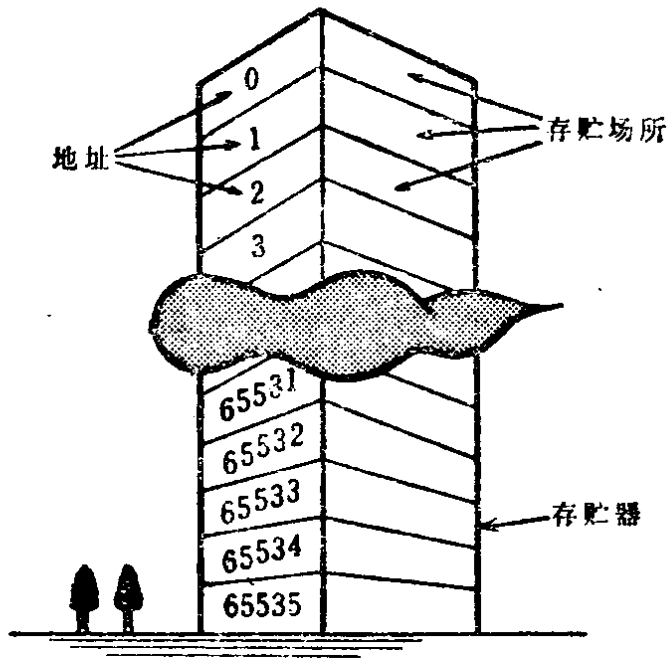


图 1.3 存贮器 (MEMORY) 和地址 (ADDRESS)

例 1.8

$$a = b + c$$

该语句的功能是把内存地址为 **b** 的存贮单元的内容(即数据)和内存地址为 **c** 的存贮单元的内容相加后，存放到内存地址为 **a** 的存贮单元里。

例 1.9

$$x1 = x1 + 5$$

该语句的功能是把内存地址为 **x1** 的存贮单元的内容加 5 后，将结果仍存放在 **x1** 存贮单元里。

(四) 有关知识之 1 ——随机数的采集

为了通过若干个抽样来测定它的母体的趋势和特性，必须不加杂人为意志地、毫无规律地、完全随机地选取标本。

采样方法之一就是使用无规则排列的**随机数**组成的**随机数表**。表 1.1 是随机数表的一部分，在这个表中数字 0，1，2，……，9 的个数几乎是相同的。为了利用这个表得到二位数的随机数，可以从这个表的任意位置开始沿着任意方向每次取两个数字，如 24，28，31，92，……。可以把这样得到的随机数作为将要选取的样本号码。

随机数表的一部分

表 1.1

2455	0032	4610	2205	0028	3072	7327
2598	9311	6116	6939	8821	4907	6896
3148	4097	0634	6982	8700	2344	4927
9214	8535	0458	0078	1466	8156	8191
3995	6759	6147	2603	7546	7451	7011
3309	9021	1400	4115	4462	7365	8617
8782	1897	6103	8091	4368	2296	3417
5241	9582	4526	7310	3125	5726	2383
3560	4781	6272	3598	1956	4776	6442
1439	2550	4189	5011	2490	3180	2801
9678	2967	4778	3887	9870	6025	7338
7009	7726	0572	9905	9831	0254	6656
2675	8816	3225	0798	1795	5697	8595
5791	9429	0205	3716	3316	7496	4955
9051	8697	7651	8572	5621	7021	2654
5566	3725	4836	3798	5995	5800	7263
1307	1525	1091	7694	5478	7561	4628
9344	3374	5317	1868	2149	4544	2103
7145	3127	6294	0932	6708	4960	8482
5088	6030	8652	4329	2255	5559	7568
6594	4738	9862	3008	8369	0334	0215
1920	9668	1017	5328	9025	8083	6606
1117	4592	2522	5405	9078	9197	5752
1373	7885	0746	9192	4305	0493	2287
5801	5199	5252	5390	7348	5388	7066

上述程序则是利用计算机能完全不受人工干预而自动产生随机数的功能，来代替人工使用随机数表制定随机数的。

二 随机数的采集(2)

——数据的贮存和比较

(一) 实例的处理

在随机数采集(1)里所谈到的程序中，对可能出现的相同的随机数没有进行处理。为了防止这种情况的发生，我们下面来改进这个程序，改进后的程序与执行结果如下：

```
10 DIM a(10)
20 FOR i=1 TO 10
30 a(i)=INT(RND*100)
40 NEXT i
50 FOR i=1 TO 9
60 FOR j=i+1 TO 10
70 IF a(i)=a(j) THEN 20
80 NEXT j
90 NEXT i
100 FOR i=1 TO 10
110 PRINT a(i);
120 NEXT i
run
24 30 31 51 5 78 49 36 98 90
OK
```

(二) 程序的要点

1. 数据的贮存