

统一书号

19106

定价：1.00元

航空高等院校教材

COBOL程序设计语言

吳偉雄 編著



航空专业教材编审组

内 容 简 介

COBOL 程序设计语言是一种用于数据处理的高级程序设计语言，也是世界上应用最广泛的程序设计语言。

全书共十一章。第一章是 *COBOL* 语言的一般规定；第二章介绍了语言中标识部分和设备部分；第三章和第四章介绍了较简单的数据结构、描述方法和最常用的指令；第五章介绍了程序设计的一般方法；第六章和第七章介绍一些较复杂的数据描述方法和指令；第八章专门讨论了文件处理方法，介绍了顺序、相对、索引三种文件；第九章讨论了提高 *COBOL* 程序质量的结构化程序编制方法；第十章介绍了 *COBOL* 排序程序的编制方法；第十一章介绍了报表打印功能。

每章配有各种程序实例并附各种思考题和练习，便于自学。

本书深入浅出，可作为高等院校管理系统工程专业、计算机专业、财会专业等学生和研究生教材；各种 *COBOL* 语言学习班的教材；也可用作实际工作者的参考书。

COBOL 程序设计语言

北京航空学院 吴伟雄 编著

责任编辑 邵景贤 王玉章

航空专业教材编审组

北京市书刊出版业营业许可证出字第074号

北京航空学院印装 内部发行

787×1092 1/19 448 千字

1982年8月第一版 1982年8月第一次印刷 印数：0,001—1,000册

统一书号：191067 定价：1.80元

前 言

随着我国四个现代化的发展,通用电子数字计算机将得到愈来愈广泛的应用。各企业、事业、科研单位、政府机关、军队等使用通用电子数字计算机的目的,主要有二类:一类是进行科学研究和工程技术的科学计算;一类是进行数据处理。如果通用电子数字计算机主要是用来进行科学计算,则它的利用率将是很低和不均衡。为了提高利用率、改善均衡性,必须用它进行数据处理工作。同时,在日常管理工作中使用通用电子数字计算机进行数据处理工作将能显著地提高管理工作的效率,以最快的速度得到所需要的各种数据及报表,为决策提供更多的素材和依据。

在数据处理中使用最普通、最广泛的程序设计语言是COBOL语言。COBOL语言是为专门的数据处理设计的,它既能用于商业,也能用于工农业、交通运输、军政部门等有关管理工作中;既能用于上级领导机关的业务管理,也能用于基层单位的有关管理中。在仓库管理、工资管理、劳动定额管理、人事管理、银行业务、情报检索以及各种报表的统计、加工、汇总、打印等工作中,使用COBOL将是最方便的。

本书着重对COBOL语言的最常用部分作一较完整的介绍并注意解决好以下几个问题:(1)本书力图处理好学习COBOL语言与编制COBOL程序脱节的矛盾,特将COBOL语言中的数据部分和过程部分都分成二段进行介绍。这样在过程部分(一)的讲授过程中,就可开始COBOL程序的编制工作。(2)本书力图克服手册式教材的组织方法,在每章中附有一定数量的COBOL程序(或部分的程序)实例,随着教学进程,程序由简单到复杂,逐步升级。(3)本书力图提高COBOL语言的易学性,对于各种术语尽可能作通俗化的解释,并配有各种实例加以说明。

全书共分十一章,其中第一章到第八章是COBOL语言的基本部分。为了提高COBOL程序编制技巧,增加了结构化程序设计这一章。此外对COBOL语言的排序和报表打印功能也作了一定的介绍。本书中的大部分程序是在罗马尼亚生产的FELIX C-256计算机上调试通过后予以简化的。

在编写过程中,承邹景贤、王玉章、马淑华、吴清秀同志提出了许多宝贵意见,朱桂芳同志协助绘图,特此表示感谢。

由于水平所限,时间仓促,本书一定会有许多错误和缺点,敬请批评指正。

编者 一九八二年一月

目 录

概论

一、COBOL 程序设计语言及其发展	1
二、COBOL 程序设计语言的优缺点	2
三、COBOL 程序运行的简介	4
四、例——一个简化的计算工资的 COBOL 程序	8
思考题	19

第一章 COBOL 语言的一般规定

一、COBOL 程序的一般构成	20
二、COBOL 语言的字符集	22
三、COBOL 字	23
四、COBOL 程序纸	32
五、卡片	36
思考题	40

第二章 标识部分 IDENTIFICATION DIVISION 及设备部

分 ENVIRONMENT DIVISION	41
一、IDENTIFICATION DIVISION 标识部分	41
二、ENVIRONMENT DIVISION 设备部分	43
思考题	46

第三章 数据部分 DATA DIVISION (一)

一、COBOL 语言数据部分的主要格式	47
二、数据结构和分层	49
三、文件名、记录名、数据名及其限定	52
四、数据项的类型及描述	55
五、数字型、字母型、字符型数据项的描述	57
六、字符编辑型数据项与数字编辑型数据项的描述	64
七、USAGE (用法) 子语句	73
八、VALUE (赋值) 子语句	77
九、数据项描述的实例	78
思考题与练习	84

~ 1 ~

第四章 过程部分PROCEDURE DIVISION (一)

一、OPEN (打开) 指令	90
二、READ (读) 指令	90
三、MOVE (传送) 指令	93
四、WRITE (写) 指令	98
五、GO TO (转移) 指令	103
六、CLOSE (关闭) 指令	106
七、STOP (停止) 指令	107
八、过程部分的实例一	107
九、ADD (加法) 指令	111
十、SUBTRACT (减法) 指令	115
十一、MULTIPLY (乘法) 指令	118
十二、DIVIDE (除法) 指令	119
十三、COMPUTE (运算) 指令	121
十四、IF (条件) 语句 (一)	123
十五、过程部分的实例二	125
思考题与练习	136

第五章 COBOL 程序的编制方法

程序的编制方法	139
思考题	145

第六章 数据部分DATA DIVISION (二)

一、OCCURS (重复) 子语句	146
二、JUSTIFIED (定位) 子语句	149
三、BLANK WHEN ZERO (当零时为空白) 子语句	150
四、REDEFINES (重新定义) 语句	151
五、RENAMES (重新命名) 语句	161
六、条件名语句	162
七、SYNCHRONIZED (同步) 子语句	163
思考题与练习	164

第七章 过程部分PROCEDURE DIVISION (二)

一、ACCEPT (接收) 指令	165
二、DISPLAY (显示) 指令	167
三、IF (条件) 语句 (二)	169
四、PERFORM (执行) 语句	172
五、EXIT (出口) 语句	185

六、程序实例	186
七、 <i>EXAMINE</i> (检测) 指令与 <i>INSPECT</i> (检查) 指令	191
思考题与练习	198
第八章 文件处理	
一、文件及其分类	200
二、文件的描述	202
三、有关文件的主要指令	204
1. <i>OPEN</i> (打开) 指令	204
2. <i>READ</i> (读) 指令	205
3. <i>START</i> (开始) 指令	205
4. <i>WRITE</i> (写) 指令	206
5. <i>REWRITE</i> (重写) 指令	206
6. <i>DELETE</i> (删去) 指令	206
7. <i>CLOSE</i> (关闭) 指令	206
四、顺序组织文件	207
五、相对组织文件	210
六、索引组织文件	219
思考题与练习	227
第九章 结构化程序设计	
一、结构化程序的主要结构	229
二、结构化程序的编制	233
三、结构化程序的举例	239
思考题与练习	244
第十章 COBOL 语言的排序	245
一、 <i>SORT</i> (排序) 指令	245
二、 <i>MERGE</i> (合并) 指令	252
第十一章 报表打印	
一、设备部分和数据部分的有关描述	253
二、报表节的描述	254
三、过程部分的有关描述	257
四、实例	258
思考题	265
附：参考书目	266

概 论

一、COBOL 程序設計語言及其发展

COBOL语言是一种适宜数据处理的高级程序设计语言。主要用于一般管理工作中。COBOL是“Common Business Oriented Language”的缩写。其中的“Common”一词是指它的通用性。一般，用高级程序设计语言编写的程序，原则上都是可以在各种计算机上运行的。但由于不同的计算机厂生产的计算机其软、硬件总有所差异，因此，用某种高级程序设计语言编制的程序不作修改，大都是无法在另一种计算机上运行的。COBOL语言也不例外。但一个已经能运行的COBOL程序转到另一种计算机使用时，只要对程序作些微小的修改，就可以顺利运行了。相对而言，有些高级程序设计语言的程序转到另一种计算机运行时程序修改工作量和复杂性要比COBOL程序大得多和麻烦得多。因此，用COBOL语言编出的COBOL程序，它的通用性是较好的。“Business oriented”的意义是“面向业务的”，明确地说，就是它的处理业务数据（资料）的能力比其科学计算的能力更强。这里的“业务数据”指的是既有数字方面的数据，也有文字方面的数据。COBOL程序能处理的，不单是数字方面的资料，而且也包括文字（主要指拉丁字母等的处理，目前汉字处理正在研究中，）方面的资料。

COBOL语言一般将其直译为“通用的商业语言”。但从其原文的意义和实际所起的作用来看，应译作“通用的面向业务语言”；或“通用管理语言”。因为它的使用范围早已远远的超出了一般商业事务范围。COBOL语言是一种主要用于一般管理工作的程序设计语言。

COBOL程序设计语言是发展最早的几种高级程序设计语言之一。1959年5月由美国政府与部分工商业者开会提出要发展一种能进行数据处理，可应用于各种型号计算机的程序设计语言。1960年4月由美国的数据系统语言会议（Conference on Data Systems Language简称CODASYL）发表了第一个正式文本，称为COBOL-60。从第一个文本开始，至今已作了多次修改，有COBOL-61、扩展的COBOL-61（1963年发表）、COBOL 1965、COBOL-68、COBOL-69、COBOL-70、COBOL-73、COBOL-76等。为了改变不同型号计算机的不相容性，美国国家标准化学会（原称ASA，现改为ANSI）和国际标准化组织（ISO）对COBOL语言进行了标准化工作。美国国家标准化学会先发表了ANSI COBOL 1968的标准文本，后又发表了ANSI COBOL 1974标准文本代替了ANSI COBOL 1968。国际标准化组织以美国1965年发表的CODASYL COBOL为蓝本制订了ISO COBOL 1972标准文本，后又以美国的ANSI COBOL 1974为蓝本，制订了ISO COBOL 1978标准文本，替代ISO COBOL 1972。这种标准化工作对推广

COBOL语言，起了很大的促进作用。

二、COBOL 程序設計語言的优缺点

大家都知道，通用电子数字计算机所采用的 程 序 设 计 语 言有多种，使用最普遍的有 *FORTRAN*（公式翻译程序设计语言 是 *FORmula TRANslator* 的缩写）、*ALGOL*（算法语言 是 *ALGOrithmic Language* 的缩写）、*APL*（程序语言 是 *A Programming Language* 的缩写）、*PL/1*（程序语言/1 是 *Programming Language /1* 的缩写）、*BASIC*（初学者通用符号指令码 是 *Beginners All-purpose Symbolic Instruction Code* 的缩写）等等。这些 程 序 设 计 语 言中的大多数是为科学研究和工程技术的科学计算设计的，虽然也可用于一般管理工作，但这些程序设计语言缺乏数据处理所需要的某些特殊功能，因此不能完全满足管理业务工作所提出的要求，如果采用编程序的方法代替这些特殊功能，将使计算机的效率大大降低，何况采用编程序的方法不一定能解决数据处理工作的全部功能要求。所以，这些程序设计语言是无法满足数据处理工作全部要求的。而在高级程序设计语言中，只有 *COBOL* 语言是专门为数据处理设计的。*PL/1* 是综合 *FORTRAN* 语言与 *COBOL* 语言而成的一种语言，因此它也能进行数据处理工作，但它的语言比 *COBOL* 更庞大。新标准的 *FORTRAN* 语言加强了数据处理的功能。

从目前来看，*COBOL* 语言仍然是世界上使用最广泛的程序设计语言。表 1 是美国某公司最近对有关程序设计语言使用情况进行抽样调查的结果：

表 1

程 序 設 計 語 言 名 稱	COBOL	RPG	FOR- TRAN	匯編語言	BASIC	PL/1	PAS- CAL	其 它
所占百分比	47.3%	14.8%	8.9%	8.4%	5.4%	4.8%	1.2%	8.9%

从表 1 中看出 *COBOL* 语言是使用最广泛的语言。

COBOL 程序设计语言所以能成为世界上使用最广泛的语言，是由于它具有以下的优点：

1. 除了能进行数字方面（0~9、小数点、正负号表示的各种数字）的处理工作外，还可进行文字方面（主要是指拉丁字母等）的资料处理工作。这样，就可用它进行某些文字信息量大的数据处理工作，如人事管理、情报检索等；

2. *COBOL* 语言比其它高级程序设计语言更接近自然语言。目前 *COBOL* 语言使用的大都是英语；它类似英语口语。它编写出的程序容易被人了解，程序的易读性较好，如

IF SALES—AMOUNT IS GREATER THAN 100.00
MULTIPLY 0.03 BY SALES—AMOUNT GIVING
COMMISSION.

其意是：假如销售总额大于100.00时，则销售总额乘以0.03，其结果放入内存中名为手续费的单元中。由此可看出，COBOL程序中的语句或指令是类似英语口语的，它在内存中存放数据的单元可直接使用通常的习惯用语（唯一的差别是，如果习惯用语是由两个以上词汇组成时，则COBOL语言在词汇间用连字符连接，如SALES—AMOUNT）。但这一优点，亦给语言为非英语的国家在使用时带来一定的困难；

3. COBOL语言规定程序中有关计算机硬件部分都集中在专门的设备部分ENVIRONMENT DIVISION内阐述。这样就使COBOL程序与具体计算机的联系只在这部分中进行。当程序需要更换到另一种计算机上运行时，不必重写程序或从程序中查找需要修改部分，只须更换这部分中的有关语句即可。由此，加强了COBOL程序的通用性，便于程序的交流和移植；

4. COBOL语言也规定将数据的描述独立成为一个数据部分DATA DIVISION。这样，它加强了数据描述的功能。它把数据分成五大类型：即字母、数字、字符（字母、数字与其它符号的混合）、数字编辑、字符编辑等，类型比其它某些程序设计语言多，更能满足数据处理工作的需要；同时它的一个数据集合（在COBOL语言中称之为组合数据项的）可以包含不同类型的数据，为数据的处理，存放等工作带来很大的灵活性和方便；

5. 可以配备专门的排序功能模块。有了这一功能模块后就能按一定的要求对数据进行有次序的排队。在数据处理工作中经常需要进行大量的排序工作，排序是数据处理要求的一项基本功能。具有这种功能模块后，将对数据处理工作带来极大的方便，可以大大节省计算机运行时间；

6. 可以配备专门的报表打印功能模块，便于大量报表的打印工作。这个功能模块是按一些最常用的报表格式设计的，采用了报表打印的功能模块后，程序的处理部分变得很简单；

7. 相对某些程序设计语言，如FORTRAN ALGOL等而言，COBOL语言的运算处理过程更接近一般管理人员（非科技人员）的逻辑思维过程，因此对一般管理人员来讲，学习COBOL语言编制程序将更为方便；

8. 适宜处理数据量大的问题，它可以大量使用磁盘等大存储量的外部设备。这些大存储量设备的使用将更有利于数据处理工作的进行。COBOL语言把各种数据资料构成顺序、相对、索引等多种形式的文件存储在各种大存储量的外部设备中，并能根据需要进行顺序、或随机、或动态的存、取和修改文件工作。COBOL程序的特点之一就是在程序中采用了较多的各种各样的文件；

9. 便于与数据库联系，可享用数据库的资料。COBOL允许和DML（数据操作语言Data Manipulation Language）相联合，使用数据库中的资料；

10. 书写格式已定型，便于放入程序库存档，查阅和相互交流。

C O B O L语言也存在一定的缺点，其缺点如下：

1. C O B O L语言比较庞大，不便记忆，用C O B O L语言编制出的程序（称之为源程序）比较长，不精炼；

2. 编制出的C O B O L程序翻译成机器语言后，质量较差。质量较差不是指它的计算精确度不够，主要是指机器指令很多，使目的程序冗长（因为一条C O B O L指令相当许多条机器指令）；

3. 运算处理的过程部分相对比较薄弱，一般只能进行加、减、乘、除和幂的运算。无法解决管理工作中所提出的需要进行复杂计算的某些问题（如某些市场预测，存货控制等问题）。

以上列举了C O B O L语言的优缺点。总的来讲，C O B O L语言的优点是主要的，所以C O B O L语言仍是目前使用最广泛的程序设计语言。

三、COBOL程序运行的简介

了解C O B O L程序在电子计算机中的一般运行过程将有助于提高学习C O B O L语言的效率。为形象化起见，现以穿孔卡片组成的一个作业为例予以说明。作业是计算机运行的一个单位。一个作业至少需要下列几组卡片才能放进电子计算机进行运算处理工作，如图1。

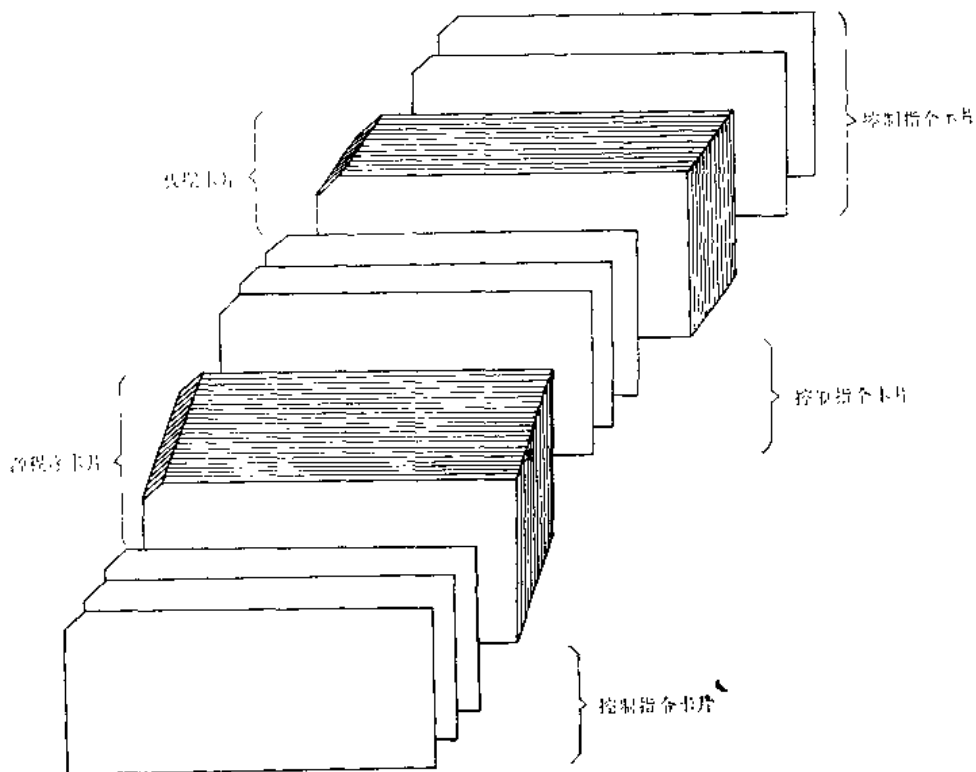


图1 卡片组成示意图

一个作业的所有卡片按图 1 的先后次序安放。所有的卡片可归纳成三类：控制指令卡片、源程序卡片和数据卡片。控制指令卡片主要是控制计算机各个运行阶段的，控制指令本身不是 *COBOL* 语言的组成部分，它是随计算机型号不同而不同的，不同牌号的计算机采用不同形式的控制指令，但其基本作用是差不多的。源程序是许多语句的组合，它准确地规定所使用和涉及的外部设备，各种输入、输出、中间使用的数据项名称、数据格式和所占用的内存单元，计算机应该执行的所有操作及先后次序等。*COBOL* 源程序就是用 *COBOL* 语言编写的源程序，所谓编 *COBOL* 程序也就是指编 *COBOL* 源程序。数据卡片是用来输入处理过程所需的原始数据。数据卡片亦不属于 *COBOL* 语言的组成部分，但它必须按照 *COBOL* 源程序中所规定的输入数据格式和各种管理工作提供的原始资料的数据进行穿孔。

源程序卡片组示意如图 2。

数据卡片组示意图如图 3。

程序编制员（简称程序员）学习 *COBOL* 语言的主要任务是编制 *COBOL* 源程序。但为了程序能在计算机上运行，还必须编制各种控制指令，在编制阶段也必须输入各种必要的原始数据，以验证程序的正确性，所以程序员必须准备出图 1 所示的全部卡片，才能放入计算机运行。由于各种计算机的 *COBOL* 语言有差异，它的控制指令格式与作用亦有差异，所以程序员必须参阅所使用计算机的有关手册，从而编制出正确的控制指令和 *COBOL* 源程序。

上述卡片组准备完毕后，就送入计算机中运行。如果卡片组是完整无误的，逻辑过程是合理的，则按图 4 运行，否则是局部运行或不运行。

联系图 1 的卡片组，对图 4 作一简单的解释。一般，卡片组第一张卡片是作业控制卡片，它告诉计算机这是一个新的作业的开始，其次的一张控制卡片是编译卡片，指明这是一个 *COBOL* 程序，需要调用 *COBOL* 编译程序进入计算机的内存（如果指明是 *FORTRAN* 程序则调用 *FORTRAN* 编译程序进入内存……）。因为计算机不能直接执行 *COBOL* 源程序，计算机只能按照机器语言进行运算处理和输入、输出。在调入编译程序后，计算机读 *COBOL* 源程序卡片，由编译程序将其翻译成机器语言，称之为目的程序。编译后的目的程序可以储存在磁盘、磁带上。接着的控制卡片是执行卡片，它把目的程序再调进计算机的内存中，这时开始输入数据并进行处理和输出，这些输入数据可以来自数据卡片。最后一张控制卡片告诉计算机本作业已结束，计算机可以执行下一个作业。

当然除了编译和运行阶段外，还可能有其它阶段，如链接装配等等。这里不赘述。

以上是一个程序在计算机中进行作业的一般过程。

图2 源程序卡片組示意图

212448	ZUN-ZIE-SANG	24950	536	03412	244	0000	1863
121437	CHANG-CHUN-SUN	16245	234	02128	244	0000	1154
121236	SAN-BAQ-ZHIE	16038	234	02363	455	0000	0535
121124	SAN-JIA-MING	26954	536	01245	834	0000	0234
111112	LI-ZEW-MIN	15300	354	01583	854	0000	0128
001001	ZHENG-E-SANG	07800	234	01000	890	0000	0000

图 3 数据卡片组示意图

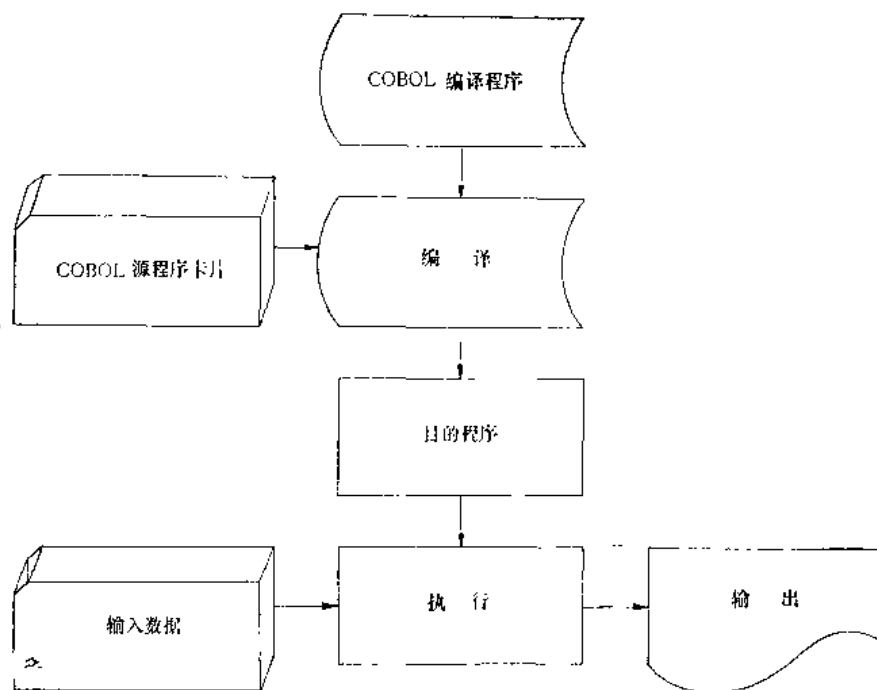


图 4 程序运行图

四、例——一个简化的计算工资的COBOL程序

在了解COBOL程序作业过程的基础上，现举一个简化的COBOL程序，这是一个计算工资的程序。其源程序如下：

```

0001 IDENTIFICATION DIVISION.
0002 PROGRAM-ID.          GONGZI.
0003 ENVIRONMENT          DIVISION.
0004 INPUT-OUTPUT         SECTION.
0005 FILE-CONTROL.
0006     SELECT KA-PIAN-WEN-JIAN ASSIGN
           TO SYSIN.
0007     SELECT DA-YIN-WEN-JIAN ASSIGN
           TO SYSOUT.
0008 DATA DIVISION.
0009 FILE SECTION.
0010 FD KA-PIAN-WEN-JIAN LABEL RECORD
           IS OMITTED.
0011 01 KA-PIAN.
0012     02 BIAN-HAO PIC 9(6).
0013     02 XING-MING PIC X(20).
0014     02 JI-BEN-GONG-ZI PIC 9(3)V99.
  
```

0 0 1 5 0 2 *FU—ZHU—GONG—ZI* *PIC* 9 *V* 9 9 .
 0 0 1 6 0 2 *JIANG—JIN* *PIC* 9 (3) *V* 9 9 .
 0 0 1 7 0 2 *BING—SHI—JIA—KOU—CHU* *PIC*
 9 9 9 *V* 9 9 .
 0 0 1 8 0 2 *TUO—ER—FEI* *PIC* 9 *V* 9 9 .
 0 0 1 9 0 2 *YI—LIAO—FEI* *PIC* 9 9 *V* 9 9 .
 0 0 2 0 0 2 *FILLER* *PIC* *X* (29) .
 0 0 2 1 *FD* *DA—YIN—WEN—JIAN* *LABEL* *RECORD*
 IS *OMITTED* .
 0 0 2 2 0 1 *DA—YIN* .
 0 0 2 3 0 2 *FILLER* *PIC* *X* (26) .
 0 0 2 4 0 2 *BIAN—HAO—D* *PIC* 9 (6) .
 0 0 2 5 0 2 *FILLER* *PIC* *XX* .
 0 0 2 6 0 2 *XING—MING—D* *PIC* *X* (20) .
 0 0 2 7 0 2 *FILLER* *PIC* *XX* .
 0 0 2 8 0 2 *J I—BEN—GONG—Z I—D* *PIC*
 9 (3) . 9 9 .
 0 0 2 9 0 2 *FILLER* *PIC* *XX* .
 0 0 3 0 0 2 *FU—ZHU—GONG—Z I—D* *PIC*
 9 (3) . 9 9 .
 0 0 3 1 0 2 *FILLER* *PIC* *XX* .
 0 0 3 2 0 2 *JIANG—JIN—D* *PIC* 9 (3) . 9 9 .
 0 0 3 3 0 2 *FILLER* *PIC* *XX* .
 0 0 3 4 0 2 *BING—SHI—J I A—KOU—CHU—D*
 PIC 9 (3) . 9 9 .
 0 0 3 5 0 2 *FILLER* *P I E* *XX* .
 0 0 3 6 0 2 *TUO—ER—FEI—D* *PIC* 9 . 9 9
 0 0 3 7 0 2 *FILLER* *PIC* *XX* .
 0 0 3 8 0 2 *Y I—L I A O—F E I—D* *PIC* 9 9 . 9 9 .

0 0 3 9 0 2 F I L L E R P I C X X .
 0 0 4 0 0 2 S H I - F A - G O N G - Z I - D P I C
 9 (4) . 9 9 .
 0 0 4 1 W O R K I N G - S T O R A G E S E C T I O N .
 0 0 4 2 7 7 J I - S H U P I C 9 9 9 V A L U E Z E R O .
 0 0 4 3 P R O C E D U R E D I V I S I O N .
 0 0 4 4 A A . O P E N I N P U T K A - P I A N - W E N - J I A N
 O U T P U T D A - Y I N - W E N - J I A N .
 0 0 4 5 M O V E S P A C E T O D A - Y I N .
 0 0 4 6 A . R E A D K A - P I A N - W E N - J I A N A T E N D
 G O T O B .
 0 0 4 7 M O V E B I A N - H A O T O B I A N - H A O - D .
 0 0 4 8 M O V E X I N G - M I N G T O X I N G - M I N G - D .
 0 0 4 9 M O V E J I - B E N - G O N G - Z I T O
 J I - B E N - G O N G - Z I - D .
 0 0 5 0 M O V E F U - Z H U - G O N G - Z I T O
 F U - Z H U - G O N G - Z I - D .
 0 0 5 1 M O V E J I A N G - J I N T O J I A N G - J I N - D
 0 0 5 2 M O V E B I N G - S H I - J I A - K O U - C H U T O
 B I N G - S H I - J I A - K O U - C H U - D .
 0 0 5 3 M O V E T U O - E R - F E I T O
 T U O - E R - F E I - D .
 0 0 5 4 M O V E Y I - L I A O - F E I T O
 Y I - L I A O - F E I - D .
 0 0 5 5 C O M P U T E S H I - F A - G O N G - Z I - D =
 J I - B E N - G O N G - Z I
 0 0 5 6 + F U - Z H U - G O N G - Z I +
 J I A N G - J I N -
 B I N G - S H I - J I A - K O U - C H U

~ 10 ~