

过程部二

一、传送语句（MOVE 语句）的较高技巧（SAMPLE5-1）

1. 各种类型数据之间的传送

非法的传送

- (1) 数值编辑项、字符编辑项、SPACE、字母数据项不能传送给数值数据项或数值编辑项。
- (2) 数值常量、ZERO，数值数据项、数值编辑项不能传送给字母数据项。
- (3) 非整数的数值数据项或数值常量不能传送给字符数据项或字符编辑数据项。

合法的传送

- (1) 接收项为字符数据项或字符编辑项，而发送项的长度大于接收项的，按“对齐原则”（一般为左对齐，当有 JUSTIFIED 子句时为右对齐），超过部分截断，如果长度小于接收项的，多余位置补空格。
- (2) 接收项是数值项或数值编辑项（初等项），可以接受数值型的数据以及内容全为数字的字符型数据。在接受数值型数据时，按小数点位置对齐，多余位补零。如果接收项是带符号的数值项，发送项的符号也要求传送。如果接收项是不带符号的数值项，按发送项的绝对值传送。如果发送项是字符型数据项，而其内容全为数字，则按无符号整数发送。整数传送时按右对齐，左截断。小数传送时，则按小数点对齐原则。
- (3) 接收项是字母型，按左对齐原则接受字母字符，多余位置补空格。但它不能接受非字母的字符

2. 组合项的传送

组合项的传送是将发送项的内容不加转换的一个字节一个字节的顺序传送到接收项。

- (1) 发送项和接收项都是组合项，而且其结构和描述均相同。则可看作将各初等项一一对应传递。
- (2) 如果发送项与接收项长度相同，但数据形式不同，则将发送项的内容原样不变的自左而右顺序的传送到接收项。它不是按初等项传送的，而将各字节内容自左而右一一传递过去。如果发送项与接收项长度不同，则按下面规则处理：左对齐，右补空格，多余位截取。

3. 对应传送(带 CORRESPONDING 子句的 MOVE 语句)

- (1) 数据名的受限和受限名的传送。

用上属层的数据名来对下属层的数据名加以“限定”，使其成为唯一。这种方法，称为数据名的受限。数据名和限定符之间用 OF 或 IN 来连接。如果限定一次不能成为唯一，则可以限定多次，数据名在受限后构成一个整体称为受限名。

- (2) 用 CORRESPONDING 子句的传送——对应传送（同名传送）

对应传送的功能是：把一个组合项中的若干项传送给另一组合项中同名的项。

一般形式为

MOVE { CORRESPONDING
CORR } 标识符 1 TO 标识符 2

说明：

- 如果两个组合项包括的项不同，则只能传送同名的项。
- 传送的两者间必须有成对的同名数据项，而且这一对中必须至少有一个项是初等项。否则不能作为对应项传递。
- 所谓同名，指的是他们有相同的全程受限。
- 带有 RENAME 子句或 REDEFINES 子句或 OCCURS 子句的数据项不予以传送。

例：02 A

03 A1 PIC 9(3)
03 A2 PIC 9(2)V99
03 A3 PIC 9V9

02 B

03 A1 PIC 99V99.
03 A3 PIC 9V9
03 A2 PIC 99V99

二、算术运算语句的较高技巧 (SAMPLE5-2)

1. 四舍五入处理 (ROUNDED 子句)

例：ADD A , B TO C ROUNDED 表示在截断后的一位四舍五入，即 $A + B + C \Rightarrow C$ 四舍五入

2. 长度溢出处理：计算结果的整数部分的长度如果比结果数据项描述所规定的整数部分长，则称长度溢出。

例：MULTIPLY A BY B GIVING C

ON SIZE ERROR DISPLAY 'SIZE ERROR'
STOP RUN

说明：在发生溢出时，在用 ON SIZE ERROR 子句时，错误的结果不存入结果数据项。

如果有几个计算结果，有的发生溢出，有的未发生溢出，未发生溢出的计算结果正常存入结果数据项，只有发生溢出的结果不存入。

当 ROUNDED 和 ON SIZE ERROR 一起使用时，先用 ROUNDED 做舍入处理，再判断是否溢出。

4. 对应项间的运算

算术运算语句中的 ADD , SUBTRACT 语句除了可以用来使两个或多个单个的数据项进行加减运算外，还可以用来使两个组合项中的对应项进行加减的计算。

一般格式 { CORRESPONDING
ADD CORR } 标识符 1 TO 标识符 2 [ROUNDED]

[; ON SIZE ERROR 强制语句]

注意：只有加法和减法才能进行有 CORR 的操作，而且对应项除了名字相同外，数据类型必须是数值型。

例：

02 A

03 A1 PIC 9(3)

03 A2 PIC 9(2)V99

03 A3 PIC 9V9

02 B

03 A1 PIC 99V99.

03 A3 PIC 9V9

03 A2 PIC 99V99

5. 除法语句中的余数子句(REMAINDER 子句)

例： DIVIDE 1.5 INTO 7 GIVING C REMAINDER D.

说明：（1）商和余数的值不仅取决于被除数和余数，还取决于数据部中对商和余数的描述

（2）如果用 ROUNDED 子句，它只对商起作用。

（3）长度溢出也只检查商的值是否溢出。

三、IF 语句的高级技巧(SAMPLE5-3)

1. IF 语句的嵌套

在嵌套的选择结构中，只能在最外层的选择结构中使用一个句点。

2. 关系表达式关系

关系表达式是以一个关系比较符将两个数据项联系起来的式子。

例： X > Y

关系比较符有两种形式：一种是接近数学符号的；一种是接近英语的。

3. 符号条件

一般形式

数据名

算术表达式

IS [NOT]

POSITIVE

NEGATIVE

ZERO

例： IF X IS POSITIVE DISPLAY X.

4. 类型条件

各个类型的数据项可以使用的条件类型

标识符类型	可用的条件类型	
	肯定型	否定型
数值型	NUMERIC	NOT NUMERIC
字母型	ALPHABETIC	NOT ALPHABETIC
字符型	NUMERIC	NOT NUMERIC
	ALPHABETIC	NOT ALPHABETIC

一般形式

标识符 IS [NOT] $\left\{ \begin{array}{l} \text{NUMERIC} \\ \text{ALPHABETIC} \end{array} \right\}$

5. 条件名条件

一个初等项只能根据某些条件取预定的几个值，或只能在一个预定的范围中取值，则这个变量称为条件变量。用来表示条件变量当前值的名字叫条件名

(1) 条件名如何说明。

条件名和条件变量要建立一定的联系，这应在数据部中说明。

例： 77 X(条件变量) PIC 9(6).

88 X1 VALUE 0 THRU 99.

88 X2 VALUE 100 THRU 999.

88 X3 VALUE 1000 THRU 9999.

其中 X1, X2, X3 为条件名。条件名用层号 88，紧跟在条件变量之后说明。经过以上在数据部的说明后，就可以在过程部中使用条件名条件。

IF X1 MOVE A TO B. (在 $0 \leq X < 100$ 时)

IF X2 MOVE A TO C.

IF X3 MOVE A TO D.

6. 复合条件

NOT , AND , OR

求复合条件的顺序：(1) 先求出每一个简单条件的值

(2) 再进行逻辑运算，先进行 NOT 运算

(3) 再进行 AND 运算

(4) 进行 OR 运算

四、字符串连接语句 (STRING 语句)

STRING 语句用来将多个非数值型的数据项的值连接起来送到一个接收数据项中，在合并过程中可以删除某些指定的字符。

STRING 语句的几种形式：

说明： A = 'HELLO '

B = 'WORLD '

C = ' ! '

1. STRING A , B , C DELIMITED BY SIZE INTO D.

则： D = 'HELLO WORLD ! '

该语句将 A 的内容从左到右传递给 D，再将 B 的内容从左到右传递给 D，之后是 C 如果有多个数据项则过程类似。

其中：DELIMITED 用来控制各个发送项的终止位置。

例如：DELIMITED BY SIZE 表示将整个发送项的字符全部送到接收项。

2. STRING A , B , C DELIMITED BY SPACE INTO D.

或 STRING A DELIMITED BY 'O'

B DELIMITED BY 'D'

C DELIMITED BY SPACE

INTO D.

则: D = 'HELLOWORLD'

或 D = 'HELLWORL'

其中: 定界符不仅可以是一个字符, 也可以是字符串, 还可以是非数值型的数据项。

3. MOVE 3 TO T

STRING A , B , C DELIMITED BY SIZE

WITH POINTER T INTO D.

则: D = ' HELLO WORLD ! '

其中: POINTER 表示逻辑指针, 其初值为 3, 如果逻辑指针的值小于 1 或大于接收项的长度, 则不能正常执行字符传送。

4. STRING A , B , C DELIMITED BY SIZE INTO D

ON OVERFLOW DISPLAY 'OVERFLOW' .

当发生溢出时显示 'OVERFLOW' 后, 程序继续执行。

STRING 语句的注意事项:

1. 接收字符串的数据项必须是初等项, 且不能有 JUST RIGHT 子句和编辑字符。
2. 指针项必须是一个整型的初等数据项
3. 在完成 STRING 语句时, 接收项中未被送入的字符位置上仍保持其原有内容, 不会设置成空格或其他字符

五、字符串分解语句 (UNSTRING 语句)

将一个发送项的数据分别传送到多个接收项中, 也就是将一个发送字符串拆成若干接收字符串, 它是 STRING 语句的逆语句。

UNSTRING 语句的几种形式:

说明: A = 'DATE PRODUCT QUANTITY '

1. UNSTRING A INTO B , C , D.

则: B = 'DATE '

C = 'PRODUCT '

D = 'QUANTITY '

该语句将 A 的内容从左到右分别存放到 B , C , D 当中, 如果有多个数据项则过程类似。

2. UNSTRING A DELIMITED BY 'T' INTO B , C.

则: B = 'DA '

C = 'E PRODUC'

DELIMITED 用来指定分解时的分界符，自左而右的传送字符，直到遇到指定的分界符。如果向右扫描发送项最右端仍未发现与分界符相同的字符，则传送以最右端的字符为截止界限。如果在 'DELIMITED BY' 后面加上 'ALL' 则表示分界符的长度是不固定的。

例：UNSTRING A DELIMITED BY ALL 'T' INTO B , C.

还可以使用多个分界符，用 'OR' 连接，表示只要满足其中之一即可。

例：UNSTRING A DELIMITED BY 'T' OR 'A' OR SPACE INTO B , C.

3. UNSTRING A DELIMITED BY 'T' INTO B COUNT IN W.

则：B = 'DA ' ,

W = 2

W 是用户定义的数据名，它的值是 'T' 字符以左的字符个数。如果数据项 A 中的第一个字符就与分界符相同则 W 为 0，其中 W 必须是整数型数据项，并且 COUNT 必须和 DELIMITED 连用。

4. 定界符存储短语 (DELIMITER)

如果有多个定界符，我们想知道送到接收项中的字符与哪个分界符相匹配，可使用 DELIMITER 短语，在传一个字符到接收项的同时，将分界符传送到相应的数据项中。

例：UNSTRING A DELIMITED BY 'T' OR ALL SPACE OR 'R'

INTO B DELIMITER IN Q

C DELIMITER IN P

D.

5. 指针短语 (POINTER)

如果希望从发送项某一指定的字符位置开始传送，可使用 POINTER 短语

例：MOVE 5 TO U.

UNSTRING A INTO C WITH POINTER U.

6. 接收项计数短语 (TALLYING)

该短语用来记录实际接受传送的接收项项数。

例：MOVE 0 TO N.

UNSTRING A INTO B , C , D TALLYING IN N.

则：N = 3

说明：N 的值是实际进行接收的接收项项数而不是字符个数。

7. 溢出短语 (OVERFLOW)

该短语的意思和 STRING 语句中的溢出短语含义相同。

UNSTRING 语句的注意事项：

1. 各短语中的先后顺序不能改变。
2. 发送项可以是组合项，也可以是字符型初等项或字符编辑型数据项
3. 接收项不能是数值编辑型数据。

4. 传送的原则依数据类型及对齐方式的不同而有所改变。

六、检测语句（INSPECT 语句）

1. INSPECT 语句可用来检查一个字符串中的字符：

- (1) 累计一个指定字符出现的次数
- (2) 用一个指定的字符去代替另一个字符
- (3) 通过指定某些字符来限制上述检查的区间

INSPECT 语句的基本格式为：

- (1) INSPECT 标识符 1 TALLYING 短语
 - (2) INSPECT 标识符 1 TALLYING 短语
 - (3) INSPECT 标识符 1 TALLYING 短语 REPLACING 短语
2. 在 INSPECT 语句中使用 BEFORE/AFTER 短语来限制被检查的区域
- 例： INSPECT A REPLACING ALL ZERO BY SPACE BEFORE ‘.’
- 或 INSPECT A REPLACING ALL ZERO BY SPACE AFTER ‘.’
3. 在 INSPECT 语句中使用 TALLYING 短语来统计满足某种条件的字符的个数
- 例： INSPECT A TALLYING N FOR CHARACTERS BEFORE ‘.’
- INSPECT A TALLYING N FOR LEADING ‘0’ BEFORE ‘.’
- INSPECT A TALLYING N FOR ALL ‘0’ BEFORE ‘.’
4. 在 INSPECT 语句中使用 REPLACING 短语用指定的字符代替另一些指定的字符
- 例： INSPECT A REPLACING ALL ‘0’ BY SPACE
- ALL ‘.’ BY ‘,’ .
- INSPECT A REPLACING FIRST ‘0’ BY SPACE BEFORE ‘.’
5. 在 INSPECT 语句中同时使用 TALLYING 和 REPLACING
- 例： INSPECT A TALLYING N FOR ALL ‘L’
- REPLACING LEADING ‘A’ BY ‘E’ AFTER ‘L’

七、转换语句（TRANSFORM 语句）

使用 TRANSFORM 语句可以使数据中的内容作某些字符转换，其一般格式为

TRANSFORM 标识符 1 CHARACTERS

FROM { 非数值常量 1
表意常量 1
标识符 2 } TO { 非数值常量 2
表意常量 1
标识符 3 }

可以通过下面的表格了解 TRANSFORM 语句的作用（T = ‘TOTAL AMOUNT’）

语 句	T 转 换 后 的 值
TRANSFORM T FROM ‘ ’ TO ‘_’	TOTAL-AMOUNT
TRANSFORM T FROM ‘AO’ TO ‘12’	T2T1L 1M2UNT
TRANSFORM T FROM ‘TUN’ TO ‘1’	1O1AL AMO11T

TRANSFORM T FROM 'A' TO SPACE	TOT L MOUNT
-------------------------------	-------------

说明：1. 我们将 T0 前面的字符称为“被转换字符”，T0 后面的字符称为“转换字符”

- 2. “转换字符”个数不能多于“被转换字符”个数
- 3. 如果“被转换字符”包含多个字符则，“转换字符”包含的字符数要么是一要么和“被转换字符”包含的字符相同
- 4. “被转换字符”中不能含有相同的字符，而“转换字符”则可以