国际图形会议译文集

A类5册

武汉工业控制计算机外部设备研究所
情　　报　　室

DATAMAX UV—1 zgrass

驻留zgrass命令和函数

1982年2月12日　　A类5册

图形／数组：
ARRAY
ARRAY.INT
ARRAY.STR
BOX
CENTER
CLEAR
CLEAR.CRT
CLEAR.WIND
DISPLAY
DISPLAY.SCREEN
DISPLAY.PAN
LINE
PATTERN
PATTERN.FILL
MMOVE

磁盘
DINIT
DLOAD
DLOAD.CLEAR
DLOAD.SET
DLOAD.ZAP
DLOOK
DPUT
DPUT.TV
DSETUP
DSETUP.RESET
DUSEMAP

输入／输出：
CONTROL P
GETDISK

图形／数组
MMOVE.UP
POINT
POINT.SNAP
POINT.PAN
SCALE
SCALE.SCR
SCALE.PAN
SCROLL
SHRINK
SNAP
STRIPE
STRIPE.STR
STRIPE.OFF
TEXT
WINDOW

输入／输出
PORT
PRINT
PRINT.FORCE
PRINT.INP
PRINT.CURSOR
PRINT.CEOL
PROMPT
PROMPT.FORCE
PUTDISK
PUTTAPE
RS232
RS232.GET
RS232.PUT
RS232.AGET
RS232.APUT

WINDOW.BOX
WINDOW.FULL
WINDOW.CENTER

磁盘：
DBAKS
DCREATE
DDELETE
DDELETE.BAK
DFETCH
DFETCH.ZAP
DGET
DGET.BAK
DGET.OR
DGET.XOR
DGET.FAST

输入／输出
RS232.RESET
TABLET
TERMINAL

数学：
ARCCOS
ARCSIN
ARCTAN
COSINE
EXP
INT
LENGTH.NUM
LN
LOG
POWER

GETTAPE
INPUT
INPUT.NAME
INPUT.STR

其它：
COMPILE
CONTROL
DELETE
DELETE.NULLS
EDIT
LOOPMAX
RESTART

程序流：
·B
·F
GOTO
IF
JUMP.ERR
RETURN
SkIP
STOP
TIMEOUT
WAIT

RS232.SGET
RS232.SPUT
RS232.BGET
RS232.BPUT

字符串处理：
ASCII
LENGTH
LPAD
STRING
STRING.NUM
SUPSTR

用户信息：
ADDRESS
ADDRESS.AR
ADDRESS.STR
ADDRESS.I
ANYARGS
CORE
HELP
STATUS
USEMAP
VERSION

SINE
SQRT
TANGENT

DATAMAX UV—I zgrass

交换命令和交换函数

1982年2月12日

图形／数组：
BUILD
ELLIPSE
CMPARA
FONT
PLACE
TXT
WRAP

磁盘
DRENAME
DZAP

错误处理：
GETERROR

其它：
ZAPI

用户信息
DEBUG
SHOW
WHATSIS

DATAMAX UV—I zgrass

常规字（BUZZWORDS），属性（IDIOSYNCRASIES）和宏系统的索引

1982年2月12日

常规字（BUZZWORDS）

（一转计算机术语）

ADDRESS 地址
ALGORITHM 算法
AND 逻辑与
ARGUMENT 自变量
ARGUMENT LIST 自变量表
ASSIGNMENT 赋值
ASSIGNMENT OPERATOR 赋值操作符
BIT 位
BYIE 字节
BYTE ARRAY 字节数组
CALL 调用
CLIPPING 剪取
COMMAND 命令
COMMENT 注释
CONCATENATION 连结
CONSTANT 常数
CONTROL CHARACTERS 控制字符
EXCLUSIVE OR 逻辑异或
EXECUTE 执行
FILES 文件
FLOATING POINT 浮点
FRAME BUFFER 帧缓冲器

FUNCTION 函数
INDEX 索引
INDIRECTION 指南
INFINITE LOOP 无限循环
INTEGER 整数
INTERRUPT 中断
ITERATION 迭代
LOGICAL OPERATOR 逻辑操作符
LOOP 循环
MEMORY 内存
NUMBER 数
NUMERIC VARIABLE 数字变量
OR 逻辑或
OVERELOW 溢出
PIXEL 象素
PLOP 换色
PRECEDENCE 优先权
PRIORITY WRITE 优先写
PUNCTUATION 标点符号
RADIAANS 弧度
RANDOW 随机
RECURSION 递归
RELATIONAL OPERATOP 关系操作符
RESOLUTION 分辨率
REVERSE PRIORITY 反优先权
SEMANTICS 语义学
SNAPPED PIX "快照"象素
STRING 字符串
STRING VARIABLE 字符串变量
SWITCH 开关
SYNTAX 语法
TRUTH TABLES 真值表
VALUE 值
VARIABLE 变量
WRAP AROUND 返转
XOR 逻辑异或

属性（IDIOSYNCRASIES）
（特殊Zgrass术语）

ABBREVIATION 缩等
CENTERING (of graphics primitives) (原图) 定中心
COLOR 颜色
COLOR MAP 颜色图
COLOR MODE 颜色方式
COORDINATES 座标
CURSOR 光标
DEVICE VARIABLE 设备变量
DISK 磁盘
DISPLAY MODES 显示方式
ERROR NUMBER 错误编号
EXPRESSION 表达式
JOYSTICK 操纵杆
LABEL 标号
LOCAL VARIABLE 局部变量
MACRO 宏
NAME 名称
NEXTLINE 下一行
OPERATOR 操作符
PANORAMA 全景
PORTS 端口
SCREEN 屏幕
SWAP COMMAND or FUNCTION 交换命令或函数
宏系统指令
NB
NC
ND

常规字(BUZZWORD),命令

函数属性(IDIOSYNCRASIES),交换命令,交换函数,

开关和秘密(ESOTERICA)的Zgrass词汇表

注意:常规字(BUZZWORD)是一般计算机术语;属性(IDIORYNCRASIE S)是属Zgrass的概念和特有的特性或者专门改进的特性;交换命令和交换函数必须首先从磁带上输入;开关是用来改进命令;秘密(ESOTERICA)是有经验的程序员使用的高级特性。

缩等(ABBREVIATION)

属性（Idiosyncrasy）

能缩写命令、函数、变量和宏名称。例如：PRINT 5 如同 PR5

当缩写名称时，如果不小心，可能会引起混乱。

例如：

```
TRY 1 = 6
TR = 2
```

这将使TRY 1等于2，因为TR是TRY 1的一种有效缩等。

为检验这一点，可执行

```
PRINT TR, TRY 1
```

绝对值

函数

参看操作符中的“！”

地址

秘密常规字（Esoteric Buzzword）

对应于内存的数据单元的号数

地址（名称）

秘密函数（Esoteric Buzzword）

返回时得到一个描述此名称的地址的一个整数。在用户内存中则此数是负数。

例如：

```
SAM = 5
PR ADDRESS( SAM )
```

返回时得到一个对应于SAM十进制数的地址。

地址（名称、数）

秘密函数（Eseteric Function）

允许读名称的首部

地址、串（串名，数）

秘密函数（Esoteric Functior）

返回时在字符串中获得对应于给定数的那一个字节。若给一个负数，可逐字地读出串的头。

地址、组（数组名、数）

秘密函数（Esoteric Fuhetioh）

返回时在数组中获得对应于给定数的那个字节。

地址.Z（交换名、数）

秘密函数（Eosteric Function）

返回时在交换模块中对应于给定数的那个字节的值。

地址名称，数，值

常规字（BUZZWORD）、命令
函数、属性（IDIOSYNCRASIES）、交换命令、
交换函数、开关、秘密（ESOTERICA的Zgrass）词汇表

注意：常规字（BUZZWORD）是一般计算机术语；属性（IDIOSYNCRASIES）是属Zgrass的概念有和特有的特性或者专门造进的特性；交换命令和交换函数必须首先从磁盘或磁带上输入；开关是用来修造命令；秘密（ESOTERICA）是有经验的程序员使用的高级特性。

ABBREVIATLON（缩写）

Idiosynerasy （属性）

能缩写命令、函数、变量和宏名称。例如：

PRINT 5 如同 PR5。

当缩写名称时，如果不小心可能会引起混乱。例如：

TRY1 = 6 TR= 2

这将使TRY1等于2，因为TR是TRY1的一种有效缩写。

为检验这一点，可执行

PRINT TR, TRY1

ABSOLUTE VALUE 绝对值

函数

参看操作符中的“!”

ADDRESS 地址

秘密常规字

对应于内存的数据单元的号数。

ADDRESS （NAME） 地址 （名称）

秘密函数

返回时获得一个描述此名称（NAME）的地址的一个整数。此数在用户内存中是负数。例如：

SAM = 5

PR ADDRESS（SAM）

返回明得到一个对应于SAM的+进制地址。

ADDRESS （NAME、NUMBER） 地址（名称、数）

秘密函数

允许读名称的首部。

ADDRESS.STR （STRNAME,NUMBER）

地址、串（串名，数）

秘密函数

返回时获得字符串（STRNAME）中对应于给定数（NUMBER）的那一个字节。若给一个负数，可逐字地读出串的首部。

ADDRESS.AR（ARNAME,NUMBER）

地址、组（数组名，数）

秘密函数

返回时获得数组（ARNAME）中对应于对定数（NUMBER）的那个字节。

ADDRESS.Z（SWAPNAME, NUMBER）

地址，Z（交换名、数）

秘密函数

返回时获得交换模块（SWAPNAME）中对应于给定数（NUMBER）那个字节的值。

ADDRESS NAME, NUMBER, VALUE

地址名称，数，值

秘密命令

把〇～255之间的值放到一个字节单元内。此单元对应于名称再加上由数(NUMBER)给定的偏值。

ADDRESS.STR STRNAME, NUMBER, VALUE

地址，串（串名，数，值）

秘密命令

把〇～255之间的值放到一个字节单元里，此字节对应于通过名称由数（NUMBER）给定的偏移量单元。这条命令正像字符串命令一样。

ADDRESS.AR ARNAME, NUMBER, VALUE

地址，组 组名，数，值

秘密命令

把〇～255之时的值放到字节单元里；此单元对应于数组名加由数（NUMBER）给定的偏移量。

ADDRESS.Z SWAPNAME, NUMBER, VALUE

地址，Z 交换名，数，值

秘密命令

把〇～255之间的值放到字节单元里；此单元对应于交换模块（SWAPNAME）加由数（NUMBER）给定的偏移量。如果你非常熟悉Z—80汇编，能使用这条命令去用目的码度变换一个交换模块。

ALGORITHM 算法

常规字

算法是用来解决一个问题的一种方法。

AND 逻辑与

常规字

“逻辑与”对位（b17）起作用，1和1“逻辑与”等于1，其它组合的“逻辑与”都等于零。用两位“逻辑与”的表：

	12	13	14	15
AND	00	01	10	11
00	00	00	00	00
01	00	01	00	01
10	00	00	10	10
11	00	01	10	11

“逻辑与”颜色方式是12—15，“逻辑与”显示方式是3，13，23……133，143。

ANYARGS( ) 取自变量表

秘密函数

若传递到宏的自变量表中没有自变量了，则返回时得到○。若自变量表中交有自变量，则返回时得到1。注意取自变量表函数（ANYARGS）是不编辑的。

```
ADDEMUP= [SUM=O
IF ANYARGS( ) = = 1, 1NPUT A;
   SUM=SUM+A; SK O
PR1NT SUM]
ADDEMUP 5, 10, 15, 20
```

ADDEMUP 是加上所有传递给它的自变量，然后打印总数，在此例中，总数量50。

ARCCOS（NUMBER） 反余弦（数）

函数

返回获得数的反正弦值。

ARCTAN（NUMBER） 反正切（数）

函数

返回获得数的反正切值。

ARGUMENT 自变量

常规字

是计算机讨论你给出命令、函数、宏中逗号之间的内容。（实际上，在第一个自变量的左边有一空格或‘(’符号，而最后一个自变量的右边有换行(NEXTL1NE)(；)或‘)’符号，而在自变量之间总是有逗号的），自变量必须是变量、数或表达式。一般来说，在命令、函数、宏中不意味着每个自变量都要出现才是合适的。

注意：自变量之间和一行结尾不允许有多余的空格。CTRL+Y将“!”符号放在标有换行（NEXTL1NE）的每一行的结尾，所以你能够说明在最后一个自变量和换行之间是否有多余的空格。

ARGUMENT LIST 自变量表常规字

是你给（传递给）命令、函数或宏的自变量表。你可用输入命令（1NPUT COM MA

ND）把自变量赋给宏中的变量。（参看1NPUT）。

秘密注释：

按名称传递给变量，当复杂表达式（A + 6 - 2）在传递时，就运算了。如果你要递传值，而此值是在单变量中（不是表达式），就使用“？”操作符。例：

```
A = 10
PR1NT A, A = 100
```

将打印出100，100。由于自变量打印之前被扫描的。

```
A = 10
PR1NT ? A, A = 100
```

将打印出10，100。这一点十分重要，即若用名称传递局部变量（没有“？”）被调用的宏不能存取调用的宏的局部变量。如果你一定要用值传递，下面是如何作的一个例子：

```
FEE = [a = 100
FOO  ? a]
FOO = [INPUT b
PRINT  b • b]
```

使用“a + o”也将强迫作数字变量计算。对于字符串用“？”（例如，？ABC）或联结一空字符串（即，ABC&〔 〕）。在全程度量中，这个问题也得到揭示。

比较：

```
TOM = [A = 100
SAM  A]
SAM = [A = 10
INPUT  B
PRINT  B • B
```

将打印100，而

```
TOM = [A = 100
SAM  A + O]
```

将打印1000。

如果你要强迫传递值，就使用“？”操作符，Egrass需要能用名称传递，所以在表达式中，能使用赋值操作符。从而某些函数（例如象TABLET）返回时可获得一个以上的值。

```
ARRAY  NAME, NUMBER
数组      名称,    数
ARRAY  NAME, N1, N2
     组数名称, , 数1, 数2
ARRAY  NAME, N1, N2, N3
数组      名称, 数1, 数2, 数3
ARRAY  NAME, N1, N2, N3, N4
数组  名称, 数1, 数2, 数3, 数4
命令
```

创建一浮点数组，数组分量由NAME（○），NAME（1），……，ANMNUEMBER—1）。注名。

1—4维数组由1—4个给定的自变量确定。例如：

```
SHOW = 〔ARRAY JANE, 200
A = O
JANE（A）= 1%100
A = A + 1
JF A<10, SK - 2
CLEAR.C
USEMAP
A = O
PR1NT “JANE（ “&A &” ）=” &IANE（A）
A = A + 1
IF A≥10, SKIP - 2〕
SHOW
```

当运行SHOW时，首先创建数组JANE，并给JANE数组的每个分量赋于一个随机数，而后生成一个用户图（USEMAP）的表，因此你能看到JANE的大小，最后打印出前10个分量。若把JANE数组改为ARRAY.INT JANE，你将注意到用户图只有原来的JA N E的一半。数组的另一个例子参看INDIRECTION。注意允许局部数组。

```
ARRAY.INT NAME, NUMBER
数组.整数  名称，数
ARRAY.1NT NAME, N1, N2
数组.整数  名称，数1，数2
ARRAY.1NT NAME, N1, N2, N3
数组.整数  名称，数1，数2，数3
ARRAY.1NT NAME, N, N2, N3, N4
数组.整数  名称，数1，数2，数3，数4
```

命令

创建一个定点数组，数组分量由NAME（O）。

NAME（1），……，NAME（NUMBER—1）备注。

1—4维数组由1—4个给定自变量确定。

注意：允许局部数组。例如：

```
ARRAY.1NT ROOTS, 10
```

将创建有10个分量数组，数组分量别是ROOTS（O），……，ROOTS（9）

```
CARS = 〔ARRAY.1NT BUICK, 100
A = O
BUICK（A）= 1%320
A = A + 1
IF A<100, SK - 2
```

```
A=O
BOX O, O, BUICJ(A), BUICK(A+1), 7
A=A+2
IF A<100, SK-2]
```

将用100个随机值填满数组BUICK，并利用它们来绘制50个方框。

```
ARRAY.INT CHECKER, C, C
```

将创建100个分量的整数数组；数组分量为CHEOKER(O, O),CHECKER(O, 1), ……, CHEOKER(9, 9)。其它例子参看INDIRECTION。

```
ARRAY.STR NAME, NUMBER
```

数组.串 名称，数

```
ARRAY.STR NAME, N1, N2
```

数组.串 名称，数1，数2

```
ARRAY.STR NAME, N1, N2, N3
```

数组.串 名称，数1，数2，数3

```
ARRAY.STR NAME, N1, N2, N3, N4
```

数组.串 名称，数1，数2，数3，数4

秘密命令

创建一个串数组，串数组分量为NAME(O)，……，NAME(NUMBER-1)。1—4维数组是由1—4给定自变量确定。为把串数组存储在磁盘或磁带上，你需要利用盘命令。

(GDSTRING/POSTRING)或带命令(GTSTRING/PTSTRING)，交模块是不可用的。

例：

```
ARRAY.STR ATHRUE, 26
ALPH=I=O
ATHRUZ(I)=ASCII(I+65)
PRINT "ATHRUZ( "&I&" )=" &ATHRUZ(I)IF(I=I+1)<26, SK—2]
```

这条宏将以字母A—Z写满串数组ATHRUZ，并打印出来。其它串数组例子参看INDIRECTION。

注意：允许局部数组。

ASCII(NUMBER) ASCII(数)

秘密函数

返回时获得对应于此数(NUMBER)即ASCII码的一个个字符串，ASCII是字符、数和标号的编码系统，参看规定值的标准ASCII表(见后)。串命令取各种字符，而返回获得对应的ASCII值。 例：

```
NUMS=〔K=48
ZEROTONINE=ZEROTONINE &ASCII(K)
IF(K=K+1)<58, SK—1
```

PRINT ZEROTONINE〕

字符O—9的ASCII码是48—57。此宏接联字符O—9，然后打印出："○1 2 3 4 5 6 7 8 9"。

控制字符、数、大写字母、小写字母和符号的ASCII值

0ONUL 17DC1 34″ 513 68D 85U
01SOH 18DC2 35# 524 69E 86V
02STX 19DC3 36$ 535 70F 87W
03ETX 20DC4 37% 546 71G 88X
04EOT 21NAK 38& 557 72H 89Y
05ENQ 22SYN 39′ 568 73I 90Z
06ACK 23ETB 40(579 74J 91〔
07BEL 24CAN 41) 58: 75K 92/
08BS 25EM 42* 59; 76L 93〕
09HT 26SUB 43+ 60< 77M 94^
10LF 27ESC 44, 61= 78N 95
11VT 28FS 45- 62> 790 96
12FF 39GS 46. 63? 80P 97a
13CR 30RS 47/ 64© 81Q 98b
14SO 31US 480 65A 82R 99c
15SI 32SP 491 66B 83S 100d
16DLE 33! 502 67C 84T 101e
102f 106j 110n 114r 118v 1222
103g 107k 111o 115s 119w 123{
104h 108e 112q 116t 120x 124¦
105i 109m 113q 117u 121y 125}

注意：抹掉键（RUB）的ASCII值是127。

下面的宏指令将打印出你所敲下键的ASCII码值。

```
GIVEASCII = 〔
IF ( A = RS232 ( O ) = = O, SKO
PRINT  A; SKIP—1〕
```

ASSIGNMENT 赋值

常规字（BuZZword）

例：A = 100

此处赋给变量A的值为100。

```
LETTERS = "ABCCEFG"
```

把字符串"ABCDEFG"赋给变量LETTERS。

```
LONGSTRING = "THIS IS A VERY VERY
LONG STRING WITH NEXTLINES AT THE
END OF EVERY LINE.NOTICE YOU
```

```
CAN HAVE NEXTLINE, COMMS, PERIODS
AND ANY OTHER PUNCTUATION EXCEPT
A DOUBLE QUOTE IN THIS CASE."
```

注意：你能把非常长的字符串赋给一个变量。

```
NULLSTRING = ""
```

一个变量能有一个空字符串作为自己的值。

```
ROOT = ( -B+SQRT ( B*B*A*C ) ) / 2
```

表达式可赋给一个变量，也能在表达式中赋值。

```
TOM = [IFA<160, BO×O, O,
       A=A+10, A, 3; SKIPO]
```

ASSIGEMENT OPERATOR 赋值操作符

常规字

赋值操作符是‘=’符号。

.B

开关

NAME.B是指在后台上NAME宏指令与其它带有.B的宏指令一再交替运行，直到碰到了（如果有的话）CTRL+C或STOP NAME为止。你将注意到当运行一个带.B的宏指令时，光标“>”仍然存在，这意味着能从键盘发送命令，执行其它宏指令或者前台宏指令(.FMACRO S)它们的执行取决它们的优先权。

例：

```
BOX  O, O, 36, 36, 1
BOX  O, 18, 4, 8, 3
SNAP APPLE,O,4,48,48
CLEAR
ANIMATE= [DISPLAY APPLE,X=X+ $X1,
                Y=Y+ $Y1,O
ANIMATE.B
```

在第一个控制杆的控制下移动APPLE（一个“快照”的（SNAPED）图形单元，直到进一步注释为正。试运行被编译的ANIMATE来体会它移动得更快，而后打入其它命令，在此命令执行的同时可见到.B宏是停顿的。

```
COMPILE ANIMATE,FASTER
FASTER.B
```

由·B宏来调用的任何宏，安好象是执行单行一样，即它不与其它.B宏交替运行。

用CTRL+RUB键，能使后台宏（.BMACRO）与正常宏（regular macro）交替运行。

BIT 位

常规字

是单个的二进制值，它为0或为1。帧上每个象素有二位。由于一位能确定两个数，二位可确定4个数，这就是为什么帧上任何一点能显示4种颜色的缘故。一个字节有8位位，而在此系统中，整数有16位，浮点数有32位。

BOX XCENTER,YCENTER,XSIZE,YSIZE,COLORMODE方框X轴心，Y轴心，X大小，Y大小，颜色方式命令

画一个尺寸为XSIZE乘YSIZE的实心矩形，其中心在XCENTER，YCENTER，并且画的方式由颜色方式（COLORMODEM）所规定（参看颜色方式的21种选择）。此命令作为函数时，若这一个位完全脱离了帧，则返回时得到—1；若使用逻辑或（OR）或者逻辑异或（XOR）方式，返回时也得到—1。若等没0则得0，若写过其它某个值得1。例：

```
BOX O,O,80,60,1
```

以颜色方式1画出中心在（O，O），具有80个象素宽，60个象素高的矩形。若你画一个整方框，放在帧上不合适时，则将会按帧的边缘来裁剪它。例如：

```
BOX 150,90,100,100,1
```

将把60×60的方框放在右上角。

BUILO NAME,XSCREEN, YSCREEN, COLORMODE,
OVERRIDE

构造 名称，

秘密交换命令

创建一个XSCREEN，乘YSCREEN的帧全景像，它们被认为是一个“附加快照”。可按给定的名称存入到4—15帧的一部分中或全部中。按照规定的颜色方式（O是清帧1是用$L1颜色填满帧，4是与帧上的位逻辑异或，好像丢掉了这位来初始化它们）全景像的大小取作XSCREEN *320，YSCREEN *201。任择的自变量“失效”（OVERRIDE）仅是规定从以前构造的一个全景象来回收整个12帧。例：

```
BUILD SAM, 3, 4, 0, 1
```

创建一个960×804象素的“附加快照”的全景像，而从以前构造的哪一个，全景像中回收空间。

```
BUILD TOM, 2, 3, 0
BUILD COPPER, 3, 2, 0
```

创设两个“附加快照”（SUper snaps），第一个是640×603象素，第二个是720×420象素，注意：第三个自变是任选的，即使无变量值也的取为O。当然你如果需要规定第4个自变量（OVERRIDE）时，你必须规定第三个自变量。

使用DISPLAY.PAN命令，PLACE命令，POINT.PAN命令，SCALE.PAN命令来存取一个“快加附照”。

注意：你不能把“附加快照”作为数组存取，显然在重新使用命令之前，你必须删去用BUILD创建的名称。在任何赋值语句中你也不能使用由BUILD创建的名称。

还要注意到，你可用DPUT、TV命令把一个的附加快照”的单独帧作为帧转贮保留起来或者能在一个磁盘上存入4～15帧的全部，而没有任何一种盘图结构。

```
DLOAD.SET
DLOAD.ZAP
```

而后用下列命令恢复它：

```
DLOAD.CLEAR
```

D LOAD

后面有用颜色方式4的构造命令（BUI LD）

BUMP STRNG, NUMBER 拼接 串，数

秘密交换命令

1 用规定的数值（NUMBER）增加到字符串最后的那个非空字符（non—null）的ASCII代码上。

例：

```
TEST = "ABCDE"
BUMP ( TEST, 2
PRiNT  TEST
```

打印出字符串"ABCDG"

注意：不象其它字符串函数，拼凑命令不能引起字符串产生新考贝。因而下面异常表现为：

```
TEST = "ABCDE"
BARB = TEST
BUMP ( TEST, 2 )
PRINT TEST, BARB
```

将两次打印"ABCDG"，由于BARB还"指到"老的但已改变的字符串。为使BARB是完全分离的字符串，将上面第二行改为：

```
BARB = TEST & [ ]
```

必要的话可改为BARB = BARB& [ ]。

BYTE 字节

常规字 （BUZZWORD）

字节是保存一个单个字符所必需的内存数量。计算机一般在每个内存单元中存一个字节。当使用用户图命令时，zgrass列出一个命过名称的数量所占据内存的字节总数。

BYTE ARRAY 字节组

常规字（BUZZWORD）

如果要存入的值在0～255之间，而你又非常短缺内存，你就可使用字符串命令作为一种存储单个字节值的方法代替存储字符的方法。字符串命令可认为是把存取串当作为一个字节数组一样。如果把"零"放入字节数组中，而试图在磁盘上存入此字符串，它将仅仅存入到"零"为此。还要注意，也不能打印此类字符串，因为有些字符会使终端关闭，清帧等等。这种节约内存的方法，只能由有经验的用户使用。

CALL 调用

常规字 （BUZZWORD）

调用是引起宏、命令或函数的执行，即此处的宏、命令或函数是按规定的名称和自变量格式给出。zgrass没有调用命令，由于规定的名称加自变量是足以调用宏、命令或者函数了。

CENTERING （of Graphis） 中心（图形的）

属性 （IOIOSXNCRASY）