

DJS—21 机编译系统 标准过程的扩充

姚卿达 肖金声
(数学力学系)

一、DJS—21 机标准过程的扩充

目前, DJS—21 计算机所采用的程序语言是 ALGOL—60, 并结合机器特点作了某些增删。ALGOL—60 中建议了九个标准函数, 包括

ABS(E) 求表达式 E 值的绝对值;

SIGN(E) 求表达式 E 值的符号;

SQRT(E) 求表达式 E 值的平方根(\sqrt{E});

SIN(E) 求 E 值的正弦;

COS(E) 求 E 值的余弦;

ARCTAN(E) 求 E 值的反正切主值;

LN(E) 求 E 值的自然对数;

EXP(E) 求 E 值的指数函数;

ENTIER(E) 求不大于 E 值的最大整数;

而且这些标准函数的名字(函数符)是保留字, 不作别用。DJS—21 机算法语言则取消“保留字”规定, 并增加了五个常用的标准函数, 即

GN3(E) 求 E 的立方根($\sqrt[3]{E}$);

TAN(E) 求 E 的正切;

ARCSIN(E) 求 E 的反正弦主值;

TOINTG(E) 对 E 四舍五入取整;

TOREAL(E) 把 E 值化为实数;

另外, 又规定了十三个标准过程, 作为输入/输出、控制、显示、访外之用, 包括:

READR

READI

```

READB
PRINT(E)
READ(A)
APRINT(A)
TPRINT(B,E,“行内符号”)
TEST(E)
PUSH(K,E)
DCXC(E1,E2,“行内符号”,E3,E4)
GUDXIN1(A,B)
XINDGU1(B,A)
JUMP(K,L)

```

规定这些标准过程，主要是为了使用的需要以及增强计算功能。

结合计算机的特点，适当增加标准函数及标准过程，这对计算人员来说往往是很需要的，同时也是丰富算法语言内容提高计算效率的一个方面。

我们在使用 DJS—21 机算法语言及其编译系统的过程中，觉得有些问题值得进一步完善和改进，例如：

1、输入/输出方面：如欲从鼓数据区读入四个实数，分别赋值给 a, b, c, d, 那么就得分用四个赋值语句来实现，即

```

a: = READR;      b: = READR;
c: = READR;      d: = READR;

```

如果要读入更多的数，而又不是数组，那么就要重复更多这样的语句。输出计算结果（快速打印 PRINT(E)）也有类似的情况，这显然是不太方便的。

2、ALGOL—60 没有定义矩阵一类的数组运算，而在数值计算中往往需要进行大量的数组运算，于是就需用较多的循环语句、条件语句及转向语句来描述算法，使程序编得冗长，而且由于编译系统没有考虑目标程序的优化，用直接法计算下标变量地址，致使数组运算时间过长，效率低。

3、内存数组与鼓数组交换时，目前只允许某个内存数组的全部元素与某个鼓数组的全部元素交换，未有部分记鼓、调鼓之功能，而且在交换中进行数组类型转换时有错，所以对于有大量数组的题目显得很不方便甚至难以进行计算。没有充分发挥鼓数组作用。

4、数值计算中常用的一些计算方法（如解线性方程组的消去法）尚未编出标准过程，计算人员都得花时间去编制和调整，影响计算效率。

为了扩充 DJS—21 机算法语言及编译系统之计算能力，增加使用上的方便，我们对编译系统的标准函数与标准过程作了初步扩充，其中包括如下几个方面：

读入一批数据——

```

READR1(K,R1,R2...,RK);

```

READI1(K,I1,I2,...,IK);

READ1(K,B1,B2,...,BK);

打印一批值——

PRINTR(K,E1,E2,...,EK);

PRINTI(K,E1,E2,...,EK);

APRINT1(K,A1,A2,...,AK);

数组运算——

SUM(A) = $\sum a_{i\dots j}$

ATOB(A,B,i);

SUMM(t,A,B,C);

STOA(B,A);

MUL(A,B,C);

MUL1(X,Y) = $\sum X_i Y_i$

部分记鼓、调鼓——

XINDGU(A,B,i);

GUDXIN(B,A,i);

常用计算过程——

GS1(A,b,X);

GJ(Ab,X,tol);

FACT(N) = N!

SMP(a,b,n,F,Y);

其他——

SONG;

DMH;

WRITE;

以上这些标准过程的意义及用法见后面“使用说明”。这些过程标识符有的按英语意义省写(如SUM,MUL tiplication,FACT orial)有的保留编译系统使用符号的习惯(如PRINTR,READI1)这些标识符未加“保留”的限制,因此仍可以用于其他目的,如作为某变量的标识符,但必须在分程序首部有所说明,而且在同一层分程序内,不能一符二义,如

```
BEGIN REAL SUM; ARRAY A[1:10];
```

```
... SUM := SUM + A[i]; ...
```

```
... PRINT (SUM(A)); ...
```

```
END
```

这里标识符SUM既是一实型简变,又是数组求和的标准过程标识符,产生不确定性(二义性)这是不允许的。

为了照顾原有习惯,算法语言中原有的各个标准过程符(如READR, PRINT, XINDGU1)照样使用,未作任何改动。

扩充的标准过程共约占一个鼓区,存貯方式有两种:一是和编译程序其他部分一起保存在0鼓,如0鼓第十五区或十四区;一是作为代码过程存貯于纸带,这只是在鼓区感到比较紧张情况下才使用,但这时对代码过程的编译要作相应的改动,使代码过程不必在分程序首部作任何说明,像使用标准函数、标准过程那样加以调用。

本扩充过程已和编译系统一起放入计算机内运行,一年来的实践表明,效果是良好的,受到广大用户欢迎。

二、DJS——21标准过程(扩充)使用说明

READR1(K,R1,R2,...,RK);

序号:0030

其中:K为整型赋值参数,R1、R2、...RK均为实型简变, $1 \leq K < 10$ (当K > 10时,如不经语法检查,仍然有效)。

功能:从鼓数据区顺序读入K个实数,分别赋给R1、R2、...、RK。

READI1(K,I1,I2,...,IK);

序号:0031

其中:K为整型赋值参数,I1、I2、...、IK均为整型简变, $1 \leq K < 10$ 。

功能:从鼓数据区顺序读入K个整数,分别赋给I1、I2、...、IK。

READ1(K,B1,B2,...,BK);

序号:0032

其中:K为整型赋值参数,B1、B2、...、BK均为数组, $1 \leq K < 10$ 。

功能:从鼓数据区顺序读入B1、B2、...、BK的全部元素。

PRINTR(K,E1,E2,...,EK);

序号:0033

其中:K为整型赋值参数,E1、E2、...、EK均为算术表达式, $1 \leq K < 10$ 。

功能:连续打印K个实数(即印出E1、E2、...、EK之值)。

PRINTI(K,E1,E2,...,EK);

序号:0034

其中:K为整型赋值参数,E1、E2、...、EK均为算术表达式, $1 \leq K < 10$ 。

功能:连续打印K个整数(即印出E1、E2、...、EK之值)。

SUM(A) = $\sum_{i=1}^k a_i$

序号:0035

其中:A为实型数组。

功能：SUM(A)是函数过程，对数组A全部元素求和。

STOA(R,A);

序号：0036

其中：R为实型赋值参数，A为实型数组。

功能：将R之值送给（赋给）A的全部元素。

SUMM(t,A,B,C);

序号：0037

其中：A、B、C均为实数组，且具有相同的维数与界偶， $t=0、1$ 。

功能：若 $t=0$ ，则 $A+B \rightarrow C$ （数组各元素求和）；若 $t=1$ ，则 $A-B \rightarrow C$ （数组相减）。

MUL(A, B, C);

序号：0038

其中：A、B、C均为二维实型数组（矩阵），而且A、B满足可乘条件，即A之列数等于B之行数。

功能：实行矩阵A与B相乘，结果送C，即 $A \cdot B \rightarrow C$ 。

$MUL1(X, Y) = \sum X_i Y_i$

序号：0039

其中：X、Y均为实型同维、同界偶数组。

功能：MUL1是函数过程，计算X与Y之内积（即对应元素两两相乘，然后求和），特例是计算向量的数量积。

ATOB(A, B, i);

序号：0030

其中：A、B为数组，i为整型赋值参数， $i \geq 1$ 。

功能：数组传送，将A的全部元素送到B中，从B中第i个元素（按B总元素计算）开始存放，（B的其余元素不变）如放不下则发出i太大之信号，转入参数错追踪。

XINDGU(A, B, i);

序号：0031

其中：A为内存数组，B为鼓数组，i为整型赋值参数， $i \geq 1$ 。

功能：部分记鼓，A的全部元素记入B中，从B的第i个元素开始存放，B其余元素不变，如放不下，则发出i太大之信号，转入参数错追踪。

GUDXIN(B,A,i);

序号：0032

其中：B为鼓数组，A为内存数组，i为整型赋值参数， $i \geq 1$ 。

功能：部分调鼓，B中从第i个开始的各元素调至内存数组A中，直到充满A为止。

功能：将计算现场写入鼓中保存，当发生意外停机时，可重调 0 区，启动 0000，当 1206 等待时，电传打入 Z，则调出现场并从该写鼓语句的后继语句继续做下去。

SMP(a,b,n,F,Y)

序号：0044

功能：用辛卜生公式求解积分 $Y = \int_a^b F(x)dx$ ，n 为分点个数，F 为实函数过程，用来算被积函数之值，使用者自编。

三、使用标准过程(扩充)的几个简单例子

例 1、《东方红》

y

BEGIN SONG END

例 2、读入与输出

y

BEGIN REAL X,Y,Z; INTEG A,B,C,D;
READR1(3,X,Y,Z); PRINTR(3,X,Y,Z);
READI1(4,A,B,C,D); PRINTI(4,A,B,C,D);
END

S

1; 2; 3; 4; 5; 6; 7;

例 3、数组运算

Y

BEGIN ARRAY A,B,C,[1:2,1:2],D,E[1:2];
STOA(1,A); STOA(2,B); STOA(3,D); STOA(4,E);
MUL(A,B,C); APRINT(C);
SUMM(0,A,B,C); APRINT(C);
SUMM(1,A,B,C); APRINT(C);
PRINT(MUL1(E,D)); PRINT(MUL1(A,B));
PRINTR(2,SUM(A),SUM(B));
ATOB(A,C,1); APRINT(C);
END

例 4、求数组全体元素之平方和、平均值，求两点间距离。

Y

BEGIN REAL Q,S,D;
ARRAY A[1:2,1:2],X,Y,Z[1:3];

```
STOA(1,A); STOA(2,X); STOA(3,Y);
Q:=MUL1(A,A); S:=SUM(A)/(2*2);
SUMM(1,Y,X,Z); D:=GN2(MUL1(Z,Z));
PRINTR(3,Q,S,D);
```

END K

JISHUAN WAN

例 5、对于给定的 X，在鼓内形成一个鼓数组 B[1:100,1:100] 即 100 × 100 矩阵，其各行元素如

$$B = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ X & X & X & \dots & X \\ X^2 & X^2 & X^2 & \dots & X^2 \\ \dots & \dots & \dots & \dots & \dots \\ X^{99} & X^{99} & X^{99} & \dots & X^{99} \end{pmatrix}$$

然后取出其第一行及最后一行快速打印出。假定 X = 0.5。

程序

Y

```
BEGIN INTEG I, REAL X;
    ARRAY A[1:100]; DRUM ARRAY B[1:100, 1:100];
X:=1; STOA(1, A); XINDGU(A, B, 1)
FOR I:=101 STEP 100 UNTIL 990 DO
    BEGIN X:=XX0.5;STOA(X,A); XINDGU(A, B,I); END;
GUDXIN(B, A, 1) APRINT(A);
GUDXIN(B, A, 9901); APRINT(A);
END K
```

JISHUAN WAN

例 6. 利用迭代法求解下列方程组：

$$\begin{cases} X_1 = 0.22X_1 + 0.02X_2 + 0.12X_3 + 0.14X_4 + 0.76; \\ X_2 = 0.02X_1 + 0.14X_2 + 0.04X_3 - 0.06X_4 + 0.08; \\ X_3 = 0.12X_1 + 0.04X_2 + 0.28X_3 + 0.08X_4 + 1.12; \\ X_4 = 0.14X_1 - 0.06X_2 + 0.08X_3 + 0.26X_4 + 0.68; \end{cases}$$

程序：

Y

```
BEGIN ARRAY A, B, C, [1:4,1:4], D, E, F [1:4];
```

```

READ1(3, A, E, F);
GS1(A, E, F); APRINT(F);
END

```

S

```

0.22; 0.02; 0.12; 0.14;
0.02; 0.14; 0.04; -0.06;
0.12; 0.04; 0.28; 0.08;
0.14; -0.06; 0.08; 0.26;
0.76; 0.08; 1.12; 0.68; 0; 0; 0; 0;

```

计算结果:

```

0 0 1   1 5 3 4 9 6 4 6 0 0
0 0 0   1 2 2 0 0 9 5 6 0 0
0 0 1   1 9 7 5 1 5 6 1 0 0
0 0 1   1 4 1 2 9 5 5 2 0 0

```

例7.利用消去法解方程组

$$\begin{aligned}
496X_1 + 33X_2 + 30X_3 + 25X_4 + 39X_5 + 29X_6 &= 25; \\
33X_1 + 33X_2 + 21X_3 + 12X_4 + 19X_5 + 16X_6 &= 22; \\
30X_1 + 21X_2 + 30X_3 + 16X_4 + 18X_5 + 13X_6 &= 20; \\
25X_1 + 12X_2 + 16X_3 + 25X_4 + 18X_5 + 12X_6 &= 20; \\
39X_1 + 19X_2 + 18X_3 + 18X_4 + 39X_5 + 20X_6 &= 23; \\
29X_1 + 16X_2 + 13X_3 + 12X_4 + 20X_5 + 20X_6 &= 20;
\end{aligned}$$

程序:

Y

```

BEGIN ARRAY A[1:6,1:7], X[1:6];
READ(A);
GJ(A, X, 10-3); APRINT(X);
END

```

S

```

4 9 6; 3 3; 3 0; 2 5; 3 9; 2 9; 2 5;
3 3; 3 3; 2 1; 1 2; 1 9; 1 6; 2 2;
3 0; 2 1; 3 0; 1 6; 1 8; 1 3; 2 0;
2 5; 1 2; 1 6; 2 5; 1 8; 1 2; 2 0;
3 9; 1 9; 1 8; 1 8; 3 9; 2 0; 3;
2 9; 1 6; 1 3; 1 2; 2 0; 2 9; 2 0;

```

计算结果如下:

```

- 0 1   - 1 8 6 2 3 9 4 8 0 0
  0 0 0   2 9 7 3 2 5 4 5 0 0
- 0 1     7 4 9 6 6 7 2 6 0 0
  0 0 0   4 3 3 4 0 9 9 8 0 0
- 0 1     9 1 2 6 7 7 3 7 0 0
  0 0 0   2 6 8 3 4 6 3 8 0 0

```

例8.在气象、水文、水产等方面的数据处理中,常常用到平隐序列的预测,即已知一个平隐序列

$$X_1, X_2, \dots, X_m$$

今要用 n 个值 $X_{m-n+1}, X_{m-n+2}, \dots, X_m$ 来预测当 $t = m+1 (i = 1, 2, \dots, l_k)$ 的 X 值。

这里我们采用最小二乘法(参见《怎样使用121机算法语言》),其中需计算平均值、中心化、算相关系数、反复解方程组、计算预测值等,为了说明使用标准过程(扩充)的某些方便之处,下面将其源程序写出,共二十六行(原书中是七十五行)。源程序后面还附了个简单数据及计算结果。

Y

```

BEGIN INTEG M, NO, N1, LO, L1, I, J, K, N, L;
READI1 (5, M, NO, N1, LO, L1);
BEGIN REAL P, S; ARRAY X(1:M), B(O:N1+L1-1);
LL1: READ (X); S := SUM(X)/M;
FOR I := 1 STEP 1 UNTIL M DO X(I) := X(I) - S;
LL2: FOR I := 0 STEP 1 UNTIL N1+L1-1 DO
BEGIN B(I) := 0; FOR K := 1 STEP 1 UNTIL M-I DO
B(I) := B(I) + X(I+K) * X(K);
B(I) := B(I)/(M-I);
END B; APRINT(B);
LL3: FOR N := NO STEP 4 UNTIL N1 DO
BEGIN ARRAY AR(1:N, 1:N+1), C(1:N);
PUSH(N, 3); FOR L := LO STEP 1 UNTIL L1 DO
BEGIN
LL4: FOR I := 1 STEP 1 UNTIL N DO
BEGIN FOR J := I STEP 1 UNTIL N DO
AR(I, J) := B(J-I); K := 0;
FOR K := K+1 WHILE KLS I DO AR(I, K) := AR(K, I)

```

```

AR [ I, N+1 ] := B [ N+L-I ];
END AR;
LL5: GJ(AR, C, 10-8); APRINT(C);
LL6: P := 0; FOR I := 1 STEP 1 UNTIL N DO
P := P + C [ I ] × (X [ M-N+I ]); P := P + S;
PRINT(P);
ENDL;
END N; END ; END

```

S

16, 4, 8, 4, 8;

0.3, 0.32, 0.33, 0.305, 0.31, 0.31, 0.32, 0.35, 0.36, 0.37, 0.36, 0.35,
0.34, 0.33, 0.32, 0.31;

计算结果:

当 n = 4 时

 $X_{m+4} = 0.32619480$ $X_{m+5} = 0.33290385$ $X_{m+6} = 0.33995273$ $X_{m+7} = 0.34468779$ $X_{m+8} = 0.34429554$

当 n = 8 时

 $X_{m+4} = 0.31520718$ $X_{m+5} = 0.31706223$ $X_{m+6} = 0.32266412$ $X_{m+7} = 0.31676216$ $X_{m+8} = 0.35778429$

例9. 计算人造卫星轨道的周长

$$Y = \int_0^{\pi/2} 4 \times 7782.5 \times \sqrt{1 - \left(\frac{9725}{7782.5} \sin(x)\right)^2} dx$$

程序

Y

```

BEGIN REAL Y;
REAL PROC F(x); VALUE X; REAL X;
F := 4 × 7782.5 × GN2 ( 1 - (972.5/7782.5 × SIN(x))2 );
Y := SMP ( 0, 3.1415927/2, 18, F, Y );
TPRINT ( 0 = 0, Y, "SMP = " );
'END'

```

结果

SMP = •48707439₁₀5;

四、标准过程扩充程序

READR1			
3 FFC	000	0000	
	000	0000	
D	-15	21C6	
	02A	2740	
E	-37	0000	
	000	0000	
F	019	0444	
	022	0888	
0000	000	0000	
	000	0000	
1	-02	0010	
	-20	0006	
2	000	0000	
	000	0000	
3	001	0000	
	020	3FFD	
4	022	0000	
	000	0000	
5	000	0001	
	-00	0004	
6	-31	0003	
	-36	0000	
7	034	3FFE	
	034	1CA2	
8	-04	0004	
	-30	0003	
9	015	1B82	
	-24	0011	
A	-02	0003	
	019	1B82	
B	-04	0003	
	-31	000C	

C	-36	0000	
	034	3FFD	
D	-04	0002	
	034	1CAA	
E	-36	0002	
	004	0000	
F	-35	0003	
	-20	000A	
0010	020	1C8F	
	000	3FFE	
1	-02	0005	
	034	1D13	

READI1			
3 FFC	000	0000	
	000	0000	
D	-15	21C6	
	02B	0740	
E	-37	0000	
	000	0000	
F	019	2666	
	033	0CCC	
0000	000	0000	
	000	0000	
1	-02	0010	
	-20	0006	
2	001	0000	
	000	0000	
3	001	0000	
	020	3FFD	
4	022	0000	
	000	0000	


```

    PRINTR
3 FFC 000 0000
      000 0000
    D -1B 1564
      -38 1A80
    E -37 0000
      000 0000
    F 019 0444
      022 0888
0000 000 0000
      000 0000
    1 -02 000E
      -20 0005
    2 001 0000
      020 3FFD
    3 022 0000
      000 0000
    4 000 0001
      -00 0003
    5 -31 0002
      -36 0000
    6 034 3FFE
      034 1CA2
    7 -04 0003
      -30 0002
    8 015 1B82
      -24 000F
    9 -02 0002
      019 1B82
    A -04 0002
      -31 000B
    B -36 0000
      034 3FFD
    C 034 1C44
      034 1BA6
    D -35 0002
    
```

```

      -20 0009
    E 020 1CF8
      000 3FFE
    F -02 0004
      034 1D13
    
```

```

    PRINTI
3 FFC 000 0000
      000 0000
    D -1B 1564
      -38 1B00
    E -37 0000
      000 0000
    F 019 2666
      033 0CCC
0000 000 0000
      000 0000
    1 -02 000E
      -20 0005
    2 001 0000
      020 3FFD
    3 022 0000
      000 0000
    4 000 0001
      -00 0003
    5 -31 0002
      -36 0000
    6 034 3FFE
      034 1CA2
    7 -04 0003
      -30 0002
    8 015 1B82
      -24 000F
    9 -02 0002
      019 1B82
    A -04 0002
    
```

- 3 1 0 0 0 B
 B - 3 6 0 0 0 0
 0 3 4 3 F F D
 C 0 3 4 1 C A 2
 0 3 4 1 B A E
 D - 3 5 0 0 0 2
 - 2 0 0 0 0 9
 E 0 2 0 1 C 8 F
 0 0 0 3 F F E
 F - 0 2 0 0 0 4
 0 3 4 1 D 1 3

APRINT1

3 F F C 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0
 D - 3 1 1 B 5 5
 - 0 9 3 8 5 D
 E - 3 6 0 0 0 0
 0 0 0 0 0 0 0 0
 F 0 1 B 0 0 0 0
 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0
 1 - 0 2 0 0 0 E
 - 0 2 0 0 0 5
 2 0 0 1 0 0 0 0
 0 2 0 3 F F D
 3 0 2 2 0 0 0 0
 0 0 0 0 0 0 0 0
 4 0 0 0 0 0 0 1
 - 0 0 0 0 0 3
 5 - 3 1 0 0 0 2
 - 3 6 0 0 0 0
 6 0 3 4 3 F F E
 0 3 4 1 C A 2
 7 - 0 4 0 0 0 3

- 3 0 0 0 0 2
 8 0 1 5 1 B 8 2
 - 2 4 0 0 0 F
 9 - 0 2 0 0 0 2
 0 1 9 1 B 8 2
 A - 0 4 0 0 0 2
 - 3 1 0 0 0 B
 B - 3 6 0 0 0 0
 0 3 4 3 F F D
 C 0 3 4 1 B B 7
 0 0 0 0 0 0 0 0
 D - 3 5 0 0 0 2
 - 2 0 0 0 0 9
 E 0 2 0 1 C 8 F
 0 0 0 3 F F F
 F - 0 2 0 0 0 4
 0 3 4 1 D 1 3

SUM

3 F F C 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 3
 D - 2 9 3 9 3 0
 0 0 0 0 0 0 0 0
 E - 3 F 3 F F F
 - 3 F 3 F F F
 F 0 3 0 0 0 0 0 0
 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0
 1 - 3 6 0 0 0 0
 0 3 4 3 F F E
 2 - 3 1 0 0 0 9
 0 2 8 1 4 1 5
 3 0 3 2 0 0 0 0
 - 3 0 0 0 0 9
 4 - 0 2 0 0 0 8

	- 36	0000		- 3F	3FFF
8	034	3FFD		F 033	0C00
	- 31	0002		000	0000
9	028	1415	0000	000	0000
	032	0000		000	0000
A	- 30	0002		1 - 20	0008
	- 36	0000		000	0000
B	034	3FFC		2 001	0000
	- 31	0003		000	0000
C	- 36	0000		3 001	0000
	034	3FFB		000	0000
D	- 31	0004		4 001	0000
	- 36	0000		000	0000
E	034	3FFE		5 001	0000
	034	1CA2		000	0000
F	117	0006		6 001	0000
	131	0010		000	0000
0010	- 02	0013		7 000	0000
	029	0008		000	0000
1	- 04	0013		8 - 36	0000
	137	0002		034	3FEF
2	002	0000		9 - 31	0002
	137	0003		028	1415
3	008	0000		A - 18	0027
	- 37	0004		032	0000
4	004	0000		B - 30	0004
	- 35	0002		- 36	0000
5	- 21	0011		C 034	3FFD
	- 20	0007		- 31	0003
				D 028	1415
				- 18	0027
	MUL			E 032	0000
3FFC	000	0000		- 30	0003
	000	0004		F - 36	0000
D	- 0D	3948		034	3FFC
	000	0000		0010 - 31	0004
E	- 3F	3FFF			

	0 3 4	3 F F D		0 0 0	0 0 0 0
0 0 1 0	- 0 4	0 0 0 4	0 0 0 0	0 0 0	0 0 0 0
	- 3 1	0 0 0 6		0 0 0	0 0 0 0
1	- 0 2	0 0 0 6	1	- 3 4	0 0 0 8
	- 1 8	0 0 0 5		0 2 0	1 C 8 F
2	0 1 9	1 B 8 2	2	- 0 0	0 0 0 3
	- 3 1	0 0 0 3		0 1 2	0 0 0 0
3	- 0 2	0 0 0 4	3	0 0 0	0 0 0 0
	0 2 8	1 4 1 5		0 0 0	0 0 0 0
4	0 3 2	0 0 0 0	4	0 0 0	0 0 0 0
	- 1 9	0 0 0 5		0 0 0	0 0 0 0
5	0 1 8	1 B 8 2	5	0 2 2	0 0 0 0
	- 3 1	0 0 0 6		0 0 0	0 0 0 0
6	- 0 2	0 0 0 6	6	0 2 2	0 0 0 0
	- 1 9	0 0 0 7		0 0 0	0 0 0 0
7	- 2 2	0 0 1 9	7	0 0 0	0 0 0 3
	- 0 2	0 0 0 3		- 0 0	0 0 0 5
8	0 3 4	1 D 1 3	8	0 0 0	0 0 0 0
	0 3 F	0 0 0 0		- 3 6	0 0 0 0
9	- 3 7	0 0 0 2	9	0 3 4	3 F F C
	0 0 2	0 0 0 0		0 3 4	1 C A 2
A	- 3 7	0 0 0 3	A	- 3 1	0 0 0 5
	0 0 4	0 0 0 0		- 3 6	0 0 0 0
B	- 3 5	0 0 0 2	B	0 3 4	3 F F D
	- 2 0	0 0 1 9		- 0 4	0 0 0 4
C	- 2 0	0 0 0 9	C	- 3 1	0 0 0 6
	0 0 0	0 0 0 0		- 0 2	0 0 0 6
			D	- 1 8	0 0 0 5
				0 1 9	1 B 8 2
			E	- 3 1	0 0 0 2
				- 0 2	0 0 0 4
			F	0 2 8	1 4 1 5
				0 3 2	0 0 0 0
3 F F C	0 0 0	0 0 0 0	0 0 1 0	- 1 9	0 0 0 5
	0 0 0	0 0 0 0		0 1 8	1 B 8 2
D	- 2 F	1 9 3 E	1	- 3 1	0 0 0 6
	0 2 A	3 F 0 0			
E	- 2 6	1 3 F F			
	- 3 F	3 F F F			
F	0 2 2	0 6 0 0			

XINDGU

- 3 6 0 0 0 0
 2 0 8 4 8 F F E
 - 3 1 0 0 0 3
 3 0 2 8 1 4 1 5
 0 3 2 0 0 0 0
 4 - 3 0 0 0 0 3
 - 3 1 0 0 1 9
 5 - 0 2 0 0 0 6
 - 1 9 0 0 1 9
 6 - 2 2 0 0 1 8
 - 0 2 0 0 0 7
 7 0 3 4 1 D 1 3
 0 3 F 0 0 0 0
 8 - 3 3 0 0 0 2
 - 2 0 0 0 0 8
 9 0 2 2 0 0 0 0
 0 0 0 0 0 0 0

GUDXIN

3 F F C 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 D - 1 7 3 9 9 6
 - 3 B 0 9 C 0
 E - 2 6 1 D F F
 - 3 F 3 F F F
 F 0 2 2 0 6 0 0
 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 1 - 3 4 0 0 0 6
 0 2 0 1 C 8 F
 2 - 0 0 0 0 0 3
 0 1 3 0 0 0 0
 3 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 4 0 2 2 0 0 0 0

0 0 0 0 0 0 1
 5 0 2 2 0 0 0 0
 0 0 0 0 0 0 0
 6 0 0 0 0 0 0 0
 - 3 6 0 0 0 0
 7 0 3 4 3 F F E
 - 3 1 0 0 0 5
 8 - 3 6 0 0 0 0
 0 3 4 3 F F D
 9 - 3 1 0 0 0 3
 0 2 8 1 4 1 5
 A 0 3 2 0 0 0 0
 - 3 0 0 0 0 3
 B - 3 6 0 0 0 0
 0 3 4 3 F F C
 C 0 3 4 1 C A 2
 - 1 8 0 0 0 5
 D - 1 9 0 0 0 4
 - 3 1 0 0 0 2
 E - 3 3 0 0 0 2
 - 2 0 0 0 0 6

FACT

3 F F C 0 0 0 0 0 0 0
 0 0 0 0 0 0 4
 D - 2 D 3 1 7 4
 0 1 0 0 0 0 0
 E - 3 7 3 F F F
 - 3 F 3 F F F
 F 0 1 8 0 0 0 0
 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 1 - 3 6 0 0 0 0
 0 3 4 3 F F E
 2 0 3 4 1 C A 2

	-04	000E		-3F	3FFF
3	-14	000D		F 033	0C00
	-24	000B		000	0000
4	002	1B82	0000	000	0000
	-04	000F		000	0000
5	002	1B84		1 -20	0005
	-04	0010		000	0000
6	-07	000F		2 001	0000
	-04	000F		000	0000
7	-02	0010		3 001	0000
	-13	000E		000	0000
8	-24	000A		4 001	0000
	-02	0010		000	0000
9	018	1B82		5 -36	0000
	-21	0005		034	3FFE
A	-02	000F		6 -31	0002
	020	1C8F		028	1415
B	-02	000C		7 032	0000
	034	1D13		-30	0002
C	000	0001		8 -36	0000
	-00	000E		034	3FFD
D	022	0000		9 -31	0003
	000	000D		028	1415
E	000	0000		A 032	0000
	000	0000		-30	0003
F	000	0000		B -36	0000
	000	0000		034	3FFC
0010	000	0000		C -31	0004
	000	0000		028	1415
				D 032	0000
				-30	0004
	GS1			E -02	0027
3FFC	000	0000		-04	0020
	000	0002		F -02	0004
D	-17	28E8		-04	0025
	000	0000		0010 -02	0002
E	-3F	3FFF			

	- 0 2	0 0 4 B		0 0 2	3 F F F
8	0 2 8	1 4 1 5	A	- 0 E	0 0 4 C
	- 3 0	0 0 4 E		- 3 6	0 0 4 E
9	- 0 2	0 0 4 A	B	0 0 3	0 0 0 0
	- 3 0	0 0 4 7		- 3 7	0 0 4 D
A	- 0 2	0 0 4 F	C	0 0 5	3 F F F
	- 0 4	0 0 4 9		- 3 7	0 0 4 E
B	- 0 2	0 0 5 0	D	0 0 4	0 0 0 0
	- 0 3	0 0 4 F		- 3 5	0 0 4 7
C	- 0 5	0 0 4 8	E	- 2 0	0 0 1 9
	- 3 7	0 0 4 E		- 0 2	0 0 4 B
D	0 1 6	0 0 0 0	F	- 3 1	0 0 4 D
	- 3 7	0 0 4 8		- 0 2	0 0 4 A
E	0 0 0	0 0 0 0	0 0 2 0	- 3 0	0 0 4 8
	- 2 4	0 0 1 3		- 0 2	0 0 4 D
F	- 0 2	0 0 4 8	1	- 1 3	0 0 4 7
	- 3 0	0 0 2 4		- 2 3	0 0 2 3
0 0 1 0	- 1 0	0 0 5 2	2	- 1 8	0 0 4 A
	- 1 A	0 0 4 E		- 0 4	0 0 4 D
1	- 3 1	0 0 4 D	3	- 2 1	0 0 2 B
	- 3 6	0 0 4 E		- 3 6	0 0 4 D
2	0 0 2	3 F F F	4	0 0 2	0 0 0 0
	- 0 4	0 0 4 C		- 1 2	0 0 4 4
3	- 3 5	0 0 4 8	5	- 0 4	0 0 4 C
	- 2 1	0 0 0 C		- 0 2	0 0 4 E
4	- 3 7	0 0 4 E	6	- 1 9	0 0 4 A
	0 0 0	0 0 0 0		- 0 4	0 0 4 E
5	- 3 5	0 0 4 E	7	- 3 7	0 0 4 E
	- 2 1	0 0 0 B		0 0 2	0 0 0 0
6	- 1 6	0 0 5 3	8	- 0 C	0 0 4 C
	- 2 4	0 0 1 8		- 3 6	0 0 4 D
7	- 0 2	0 0 0 1	9	0 0 8	0 0 0 0
	0 3 4	1 D 1 3		- 3 7	0 0 4 D
8	- 0 2	0 0 4 7	A	0 0 4	0 0 0 0
	- 0 4	0 0 4 E		- 3 5	0 0 4 B
9	- 3 6	0 0 4 D	B	- 2 0	0 0 2 7

	-35	0049		032	0000
C	-21	001F		E -30	004B
	-35	004B		-30	0054
D	-21	0006		F -36	0000
	-02	004A		034	3FFD
E	028	0423	0040	-31	0054
	-04	004E		-36	0000
F	029	0001		1 034	3FFC
	-04	004C		034	1C44
0030	-02	004B		2 -04	0053
	-31	004E		-20	0039
1	-37	004C		3 000	0000
	000	0000		000	0001
2	-02	0051		4 000	2000
	-37	004E		000	0000
3	016	0000		5 001	0000
	-25	0032		000	0000
4	-02	004E		6 022	0000
	-19	004C		000	0002
5	032	0000		7 001	0000
	-37	004B		000	0000
6	004	0000		8 001	0000
	-37	0054		000	3FFF
7	004	0000		9 001	0000
	-35	004F		000	3FFF
8	-20	0030		A 001	0000
	020	1C8F		000	0000
9	000	0000		B 001	0000
	-02	0045		000	0000
A	-04	004B		C 001	0000
	-04	0054		000	0000
B	-36	0000		D 001	0000
	034	3FFE		000	0000
C	-31	004B		E 001	0000
	028	1415		000	0000
D	-18	0046		F 001	0000

	000	014A		000	014A	
A	000	0013		C	000	0019
	000	00A5			000	00F6
B	000	0011		D	000	001C
	000	00B9			000	00DC
C	000	0019		E	000	001E
	000	01EC			000	0068
D	000	001C		F	000	0022
	000	00DC			000	005C
E	000	001C	0030	000	000	0026
	000	006E			000	00A4
F	000	0022		1	000	0013
	000	005C			000	014A
0020	000	0019		2	000	0019
	000	01EC			000	00F6
1	000	0013		3	000	0017
	000	014A			000	008A
2	000	0013		4	000	0019
	000	014A			000	007B
3	000	0011		5	000	001C
	000	00B9			000	00DC
4	000	000E		6	000	001C
	000	00DC			000	006E
5	000	0011		7	000	0022
	000	00B9			000	005C
6	000	0013		8	000	0019
	000	00A5			000	007B
7	000	001C		9	000	0017
	000	00DC			000	008A
8	000	001C		A	000	0019
	000	006E			000	007B
9	000	0022		B	000	001C
	000	005C			000	006E
A	000	0019		C	000	0019
	000	01EC			000	007B
B	000	0013		D	000	001C

	000	006E		000	0000
E	000	001E	0	000	0000
	000	0068		136	0000
F	000	0022	1	034	3554
	000	005C		034	1244
0040	000	0026	2	104	0002
	000	0148		136	0000
			3	034	3553
				034	1244
	SMP		4	104	0003
3FFC	000	0000		136	0000
	000	0000	5	034	3552
D	129	0368		034	1202
	000	0000	0010	131	0008
E	137	1355		006	0000
	135	3555	1	104	0004
F	011	0744		136	0000
	000	0000	2	034	3550
0000	000	0000		104	0005
	000	0000	3	136	0000
1	134	0000		034	3550
	020	1285	4	104	0006
2	000	0000		102	0003
	000	0000	5	109	0002
3	000	0000		004	1185
	000	0000	6	104	0004
4	000	0000		104	0007
	000	0000	7	120	0010
5	000	0000		102	0002
	000	0000	8	004	1430
6	000	0000		002	1101
	000	0000	9	000	0000
7	000	0000		121	0017
	000	0000	0	102	0005
8	001	0000		034	1221
	000	0000	1	104	0009
9	000	0000			

	0 0 0	0 0 0 0	4	0 0 0	0 0 0 0
2	1 2 0	0 0 1 5		1 2 1	0 0 1 2
	1 0 2	0 0 0 3	5	1 0 2	0 0 0 5
3	0 0 4	1 4 3 0		0 3 4	1 2 2 1
	0 0 2	1 1 0 1			

参 考 资 料

- [1] P.Neur: Report on the Algorithmic Language ALGOL 60.
 "NUMERISCHE MATHEMATIC", 1960.6.
- [2] 八三〇厂程序组: DJS—21机编译系统。“数学的实践与认识” 1973.4.