

000001

```

THEN WRITE(F,= <UNDEF,>:=11); TRUE);
ELSE IF BVAL THEN WRITE(F,= FALSE);
ELSE WRITE(F,= :=12);
WRITE(F,= :=12);
END;
6: (* ALFA *)
7: (* POINTER *)
    WRITE(F,NAME:=11,= :=12);
    BEGIN
        IF PVAL=NIL THEN WRITE(F,= NIL);
        ELSE WRITE(F,= :=5,IVAL: 6 OCT);
        WRITE(F,= :=12);
    END;
END;
IF P THEN WRITELN(F) ELSE WRITE(F,= );
P := NOT P
END;
IF P THEN WRITELN(F);
STACKOF := FALSE
END;
UNTIL B2 = B5
END
END (* PMD *);

(*$X0,T-,U+,P-,E+ WIRTH/JENSEN *)
FUNCTION TEN(E: INTEGER): REAL; (* = 10**E, 0<E<322 *)
VAR I: INTEGER; T: REAL;
BEGIN T := 0; T := 1.0;
REPEAT IF ODC(E) THEN
CASE I OF
 0: T := T * 1.0E1;
 1: T := T * 1.0E2;
 2: T := T * 1.0E4;
 3: T := T * 1.0E8;
 4: T := T * 1.0E16;
 5: T := T * 1.0E32;
 6: T := T * 1.0E64;
 7: T := T * 1.0E128;
 8: T := T * 1.0E256;
END;
E := E DIV 2; I := I+1
UNTIL E = 0;
TEN := T
END (* TEN *);

(*$T-,X0,E+,U+,P- READ REAL NUMBERS. WIRTH/JENSEN*)
PROCEDURE RDR(VAR F: TEXT; VAR X: REAL);
(* READ REAL NUMBERS IN FREE FORMATE *)
CONST T48 = 281474976710656;
LIMIT = 56294995342131;
Z = 27; (* ORD(E0E) *)
LIM1 = 322; (* MAXIMUM EXPONENT *)
LIM2 = -292; (* MINIMUM EXPONENT *)
TYPE POSINT = 0..323;
VAR CH: CHAR; Y: REAL; A,I,E: INTEGER;
S,SS: BOLEAN; (* SIGNS *)

FUNCTION TEN(E: POSINT): REAL; (* = 10**E, 0<E<322 *) EXTERN;
BEGIN
IF EOF(F) THEN
BEGIN MESSAGE(*TRIED TO READ PAST EOS/EOF*); HALT;
END;
(*SKIP LEADING BLANKS*)
WHILE (F=:= ) AND (NOT EOF(F)) DO GET(F);
IF NOT EOF(F) THEN
BEGIN
CH := F;
IF CH = +- THEN
BEGIN S := TRUE; GET(F); CH := F;
END ELSE
BEGIN S := FALSE;
IF CH = := THEN
BEGIN GET(F); CH := F;
END;
END;
IF NOT(CH IN {E0E..E9E}) THEN
BEGIN MESSAGE(*DIGIT EXPECTED*); HALT;
END;
A := 0; E := 0;
REPEAT IF A < LIMIT THEN A := 10*A + ORD(CH)-Z ELSE E := E+1;

```

000003

```

GET(F); CH := F↑
UNTIL NOT(CH IN [E0E..E9E]);
IF CH = E THEN
BEGIN (* READ FRACTION *) GET(F); CH := F↑;
  WHILE CH IN [E0E..E9E] DO
    BEGIN IF A < LIMIT THEN
      BEGIN A := 10*A + ORD(CH)-2; E := E-1
      END;
    GET(F); CH := F↑
  END;
END;
IF CH = EEE THEN
BEGIN (* READ SCALE FACTOR *) GET(F); CH := F↑;
  I := 0;
  IF CH = E-E THEN
    BEGIN SS := TRUE; GET(F); CH := F↑
  END ELSE
    BEGIN SS := FALSE; IF CH = E+E THEN
      BEGIN GET(F); CH := F↑
    END;
  END;
  IF CH IN [E0E..E9E] THEN
    BEGIN I := ORD(CH) - 2; GET(F); CH := F↑;
      WHILE CH IN [E0E..E9E] DO
        BEGIN IF I < LIMIT THEN I := 10*I + ORD(CH) - 2;
          GET(F); CH := F↑
        END;
    END;
  END ELSE
    BEGIN MESSAGE(**DIGIT EXPECTED**); HALT END;
  IF SS THEN E := E-I ELSE E := E+I;
END;
IF E < LIM2 THEN
  BEGIN A := 0; E := 0
  END ELSE
    IF E > LIM1 THEN
      BEGIN MESSAGE(**NUMBER TOO LARGE**); HALT END;
    (* 0 < A < 2**49 *)
    IF A >= T48 THEN Y := ((A+1) DIV 2) * 2.0 ELSE Y := A;
    IF S THEN Y := -Y;
    IF E < 0 THEN X := Y/TEN(-E) ELSE
      IF E > 0 THEN X := Y*TEN(E) ELSE X := Y;
  END;
END (* RDR *);

```

(*\$T-,X0,E+,U+,P- WRITE REAL NUMBERS. WIRTH/JENSEN*)

```

PROCEDURE WRE(VAR F:TEXT; X: REAL; N: INTEGER);
  (* WRITE REAL NUMBER X IN N CHARACTERS *)
  CONST T48 = 281474976719656; (* 2**48 *)
  Z = 27; (* ORD(E0E) *)
  TYPE POSINT = 0..323;
  VAR C,D,E,E0,E1,E2,I: INTEGER;

  FUNCTION TEN(E: POSINT): REAL; (* 10**E, 0<E<322 *)
    EXTERN;
  BEGIN (* AT LEAST 10 CHARACTERS NEEDED: B+9.9E+999 *)
    IF NOT UNDEFINED(X) THEN X := X+0.0;
    IF UNDEFINED(X) THEN
      BEGIN REPEAT F↑ := E; PUT(F); N := N - 1 UNTIL N <= 1;
        F↑ := E; PUT(F)
      END ELSE
        IF X=0 THEN
          BEGIN REPEAT F↑ := E; PUT(F); N := N - 1 UNTIL N <= 1;
            F↑ := E0; PUT(F)
          END ELSE
            BEGIN
              IF N <= 10 THEN N := 3 ELSE N := N-7;
              REPEAT F↑ := E; PUT(F); N := N-1
              UNTIL N <= 15;
              BEGIN (* TEST SIGN, THEN OBTAIN EXPONENT *)
                IF X < 0 THEN
                  BEGIN F↑ := E-E; PUT(F); X := -X
                  END ELSE BEGIN F↑ := E-E; PUT(F) END;
                E := EXP0(X);
                IF E >= 0 THEN
                  BEGIN E := E*77 DIV 256 + 1; X := X/TEN(E);
                    IF X >= 1.0 THEN
                      BEGIN X := X/10.0; E := E + 1
                      END
                  END ELSE
                    BEGIN E := (E+1)*77 DIV 256; X := TEN(-E)*X;
                  END;
                END;
              END;
            END;
          END;
        END;
      END;
    END;
  END;

```

```
IF X < 0.1 THEN
BEGIN X := 10.0*X; E := E - 1
END
```

000004

```
END;
(* 0.1 <= X < 1.0 *)
CASE N OF
 2: X := X+0.5E-2;
 3: X := X+0.5E-3;
 4: X := X+0.5E-4;
 5: X := X+0.5E-5;
 6: X := X+0.5E-6;
 7: X := X+0.5E-7;
 8: X := X+0.5E-8;
 9: X := X+0.5E-9;
10: X := X+0.5E-10;
11: X := X+0.5E-11;
12: X := X+0.5E-12;
13: X := X+0.5E-13;
14: X := X+0.5E-14;
15: X := X+0.5E-15;
END;
IF X >= 1.0 THEN
BEGIN X := X + 0.1; E := E+1
END;
C := TRUNC(X, 48);
C := 10*C; D := C DIV T48;
F^ := CHR(D+Z); PUT(F);
F^ := EEE; PUT(F);
FOR I := 2 TO N DO
BEGIN C := (C - D*T48) * 10; D := C DIV T48;
F^ := CHR(D+Z); PUT(F);
END;
F^ := EEE; PUT(F); E := E-1;
IF E < 0 THEN
BEGIN F^ := EEE; PUT(F); E := -E;
END ELSE BEGIN F^ := EEE; PUT(F); END;
E1 := E * 205 DIV 2048; E2 := E - 10 * E1;
E0 := E1 * 205 DIV 2048; E1 := E1 - 10 * E0;
F^ := CHR(E0+Z); PUT(F);
F^ := CHR(E1+Z); PUT(F);
F^ := CHR(E2+Z); PUT(F)
END
END (* WRE *) ;
```

(*\$T-,X0,E+,U+,P- WRITE INTEGERS. WIRTH/JENSEN*)

```
PROCEDURE WRI(VAR F: TEXT; X: INTEGER);
CONST Z = 27; (* ORD('0') *)
      LIMIT = 281474976710656; (* 2**48 *)
VAR I, Q: INTEGER; S: BOOLEAN;
D: ARRAY [1..16] OF CHAR;
BEGIN (* WRITE X IN DECIMAL FORM WITH N CHARACTERS *)
  IF X >= 0 THEN S := FALSE ELSE
    BEGIN S := TRUE; X := -X
    END;
  IF X >= LIMIT THEN
    BEGIN WHILE N > 1 DO
      BEGIN F^ := EEE; PUT(F); N := N-1
      END;
      F^ := EEE; PUT(F)
    END ELSE
    BEGIN (* DECIMAL DECOMPOSITION *)
      I := 0;
      REPEAT I := I+1; Q := X DIV 10;
      D[I] := CHR((X - 10*Q) + 2); X := Q
      UNTIL X = 0;
      IF S THEN BEGIN I := I+1; D[I] := EEE END;
      WHILE N > I DO
        BEGIN F^ := EEE; PUT(F); N := N-1
        END;
      REPEAT F^ := D[I]; PUT(F); I := I-1
      UNTIL I = 0
    END
  END (* WRI *) ;
```

(*\$T-,X0,E+,U+,P- READ INTEGERS. WIRTH/JENSEN*)

```
PROCEDURE ROI(VAR F: TEXT; VAR X: INTEGER);
(* READ INTEGER IN EFREE FORMAT *)
CONST Z = 27; (* ORD('0') *)
      LIM1 = 28147497671066; (* 2**48 / 10 +1 *)
```

LIM2 = 281474976710655; (* 2**48 - 1 *)

```

  VAR CH: CHAR;
  A: INTEGER; S: BOOLEAN;
BEGIN
  IF EOF(F) THEN
    BEGIN MESSAGE('**TRIED TO READ PAST EOS/EOF'); HALT;
  END;
  (*SKIP LEADING BLANKS*)
  WHILE (F <= ' ') AND (NOT EOF(F)) DO GET(F);
  IF NOT EOF(F) THEN
    BEGIN
      CH := F;
      IF CH = '-' THEN
        BEGIN S := TRUE; GET(F); CH := F;
      END ELSE
        BEGIN S := FALSE;
        IF CH = '+' THEN
          BEGIN GET(F); CH := F;
        END;
      END;
      IF NOT (CH IN ['0'..'9']) THEN
        BEGIN MESSAGE('**DIGIT EXPECTED'); HALT;
      END;
      A := 0;
      REPEAT IF A < LIM1 THEN A := 10*A + ORD(CH)-Z
      ELSE
        BEGIN
          MESSAGE('**INTEGER TOO LARGE'); HALT;
        END;
      GET(F); CH := F;
    UNTIL NOT (CH IN ['0'..'9']);
    IF A >= LIM2 THEN BEGIN MESSAGE('**INTEGER TOO LARGE'); HALT;
    END;
    IF S THEN X := -A ELSE X := A
  END (* RDI *);
(*$T-,X0,E+,U+,P- WRITE BOOLEANS. JENSEN*)

PROCEDURE WRB(VAR F: TEXT; B: BOOLEAN; N: INTEGER);
BEGIN
  WHILE N>5 DO
    BEGIN N := N-1; F := ' ' PUT(F)
    END;
  IF B THEN
    BEGIN F := 'T' PUT(F);
    F := 'R' PUT(F);
    F := 'U' PUT(F);
    F := 'E' PUT(F);
    F := 'E' PUT(F);
  END ELSE
    BEGIN F := 'F' PUT(F);
    F := 'A' PUT(F);
    F := 'L' PUT(F);
    F := 'S' PUT(F);
    F := 'E' PUT(F);
  END;
END (* WRB *);

(*$T-,X0,E+,U+,P- WRITE FIXED-POINT NUMBERS. WIRTH/JENSEN*)

PROCEDURE WRF(VAR F: TEXT; X: REAL; M,N: INTEGER);
(* WRITE REAL NUMBER X IN FIXED-POINT FORMAT;
  M CHARACTERS, N AFTER THE '.')
CONST T48 = 281474976710656; (* 2**48 *)
Z = 27; (* ORD('0') *)
TYPE POSINT = 0..323;
VAR C,D,E,R,K2,K3,I: INTEGER;
SIGN: CHAR;
FUNCTION TEN(E: POSINT): REAL; (* 10**E, 0<E<322 *)
  EXTERN;
BEGIN
  IF UNDEFINED(X) THEN
    BEGIN REPEAT F := ' ' PUT(F); M := M - 1 UNTIL M <= 1;
    F := 'E'; PUT(F)
    END ELSE
    BEGIN
      (* X = 0: A SPECIAL CASE *)

```

```

IF X=0 THEN REPEAT F↑ := EEE; PUT(F); M := M-1 UNTIL M<=1;
F↑ := EOE; PUT(F)
END ELSE
BEGIN (* CHECK SIGN AND DETERMINE SCALE FACTOR *)
IF X<0 THEN
BEGIN SIGN := EEE; X := -X
END ELSE SIGN := EEE;
E := EXP0(X);
IF E >= 0 THEN
BEGIN E := E*77 DIV 256 + 1; X := X/TEN(E);
IF X >= 1.0 THEN
BEGIN X := X/10.0; E := E + 1
END
END ELSE
BEGIN E := (E+1)*77 DIV 256; X := TEN(-E)*X;
IF X < 0.1 THEN
BEGIN X := 10.0*X; E := E - 1
END
END;
(* 0.1 <= X < 1/0 *)
IF N<0 THEN N := 0; R := N+E;
IF R >= 0 THEN
IF R < 16 THEN
CASE R OF
0.. X := X+0.5;
1.. X := X+0.1;
2.. X := X+0.2;
3.. X := X+0.3;
4.. X := X+0.4;
5.. X := X+0.5;
6.. X := X+0.6;
7.. X := X+0.7;
8.. X := X+0.8;
9.. X := X+0.9;
10.. X := X+0.0;
11.. X := X+0.1;
12.. X := X+0.2;
13.. X := X+0.3;
14.. X := X+0.4;
15.. X := X+0.5;
END;
IF X >= 1.0 THEN
BEGIN X := X * 0.1; E := E+1
END;
IF E>0 THEN
BEGIN M := M-N-E-2; K2 := E; K3 := 0
END ELSE
BEGIN M := M-N-3; K2 := 0;
IF N+E >= 0 THEN K3 := -E ELSE K3 := N;
N := N-K3
END;
(* M BLANKS, SIGN, K2 DIGITS, E..E, K3 ZEROES, N DIGITS *)
WHILE M > 0 DO
BEGIN F↑ := EEE; PUT(F); M := M - 1 END;
F↑ := SIGN; PUT(F);
C := TRUNC(X,48); D := 0;
IF K2=0 THEN BEGIN F↑ := EOE; PUT(F) END ELSE
REPEAT C := (C - D*T48) * 10;
D := C DIV T48;
F↑ := CHR(D+Z); PUT(F); K2 := K2-1
UNTIL K2=0;
F↑ := EEE; PUT(F);
WHILE K3 > 0 DO
BEGIN F↑ := EOE; PUT(F); K3 := K3-1
END;
WHILE K<>0 DO
BEGIN C := (C - D*T48) * 10;
D := C DIV T48;
F↑ := CHR(D+Z); PUT(F); N := N-1
END;
END;
* WR *);

```

END:
F↑ t= CH: PUT(F)
END (* WRC *) ;
BEGIN (* DUMMY *) END.

000007

SYSIO
SYSIO
SYSIO
SYSIO
SYSIO

41
41
41
41
41

RETURN TO JWD

10,58,22..000011 PAGES PRINT. 000501 LINES PRINT. FOR \$ 000.56 AT RG2.