

```

//JCRI LISP JOB (205203,3,0,2),*REYNOLDS COL 1 ROW 3',MSGLEVEL=1
REYNOLDS J 05203AMD344X51430-01145050
//JOB LIB DD DSN= C145.B17648.LISP.OBJECT,DISP=SHR
// EXEC PGM=LISP
//LISPOUT DD SYSOUT=A,DCB=BLKSIZE=798
//GEDANK DD UNIT=CELL,VOL=SER=CELLBE,
// DSN= C145.B05203.GEDANKEN.APRIL29A,
// DISP=SHR
//LISP IN DD *
OPEN (GEDANK SYSFILE INPUT)
RESTORE (GEDANK)
CLOSE (GEDANK)
EXCISE (NIL)
PRBUFFER (T)
GEDANKEN()
NIL IS ATOM();
PRNV IS #X (N IS REF X LL; PRINT " ";
PRINT X LL;
LOOP: IF GREATER(N, X UL) THEN GOTO DONE ELSE ();
PRINT X VAL N; N := INC N; GOTO LOOP;
DONE: PRINT X UL);
PRNVR IS #X (N IS REF X LL; PRINT " ";
PRINT X LL;
LOOP: IF GREATER(N, X UL) THEN GOTO DONE ELSE ();
PRINT VAL X VAL N; N := INC N; GOTO LOOP;
DONE: PRINT X UL);
PVECTOR IS #(L,U,F) (L IS COERCE L; U IS COERCE U; F IS COERCE F;
#I (I IS COERCE I; IF I = LL THEN L ELSE IF I = UL THEN U
ELSE F I));
LIBRARY$
GEDANKEN ()
M IS (M1 IS VECTOR(1, 3, # I VECTOR(1, 3, # J REF 0));
#(I, J) (M1 I) J);
MT IS #(I, J) M(J, I); MD IS # I M(I, I);
S IS (S1 IS VECTOR(1, 3, # I VECTOR(1, I, # J REF 0));
#(I, J) IF NOT GREATER(J, I) THEN (S1 I) J ELSE (S1 J) I);
S(1,3) := M(1,3) := 1;
S(2,1) := MT(1,2) := 2;
S(3,2) := MT(2,3) := 3;
M(3,1) := 4;
MT(2,1) := 5;
MT(3,2) := 6;
S(2,2) := M(2,2) := 7;
S(3,3) := MD(3) := 8;
VECTOR(1,3, #I VECTOR(1,3, #J
( TERPRI(); PRINT I; PRINT J; PRINT VAL M(I,J); PRINT VAL MT(I,J);
PRINT VAL S(I,J); PRINT VAL MD I)););
OS
GEDANKEN ()
READ IS READCHAR;
MAKERLIST ISR #()
(B IS REF 0; # I
(IF B = 0 THEN B := (READ(), MAKERLIST()) ELSE (); B I));
(X IS MAKERLIST());
X1 IS X 1; X2 IS X 2;
PRNV(X1,X 1,X2 1,X 1,(X 2) 1,X2 1,((X 2) 2) 1,X 1, (X 2) 1, ((X 2)2)1)
);
OSABC
GEDANKEN ()

```

PSEUDOCOPY IS # V

(CCL IS REF NIL;

SEARCHCL ISR #(X, I, F, G) IF X = NIL THEN G() ELSE IF (X 1) 1 = I THEN F (X 1) 2

ELSE SEARCHCL(X 2, I, F, G);

* I (I IS COERCE I;

IF I = LL THEN V LL ELSE IF I = UL THEN V UL

ELSE IF NOT ISINTEGER I

OR GREATER(V LL, I) OR GREATER(I, V UL)

THEN GOTO ERROR

ELSE IMPREF(

X SEARCHCL(VAL CL, I, # R NCSET(R, X),

#() CL := ((I, NCREF X), VAL CL)),

#() SEARCHCL(VAL CL, I, VAL, #() VAL V I)))));

V IS (REF 1, REF 2, REF 3);

C IS PSEUDOCOPY V;

PRNVR C;

C 1 := 5; C 2 := 6;

PRNVR C;

C 1 := 7;

PRNVR C;

PRNVR V;

OS

GEDANKEN ()

COMPILE IS #A (A 1; (GOTO L; M: A 3; GOTO N; L: A 2; GOTO M); N: A 4);

ASSEMBLE ISR #A (X IS PRINT A()); IF X = 4 THEN () ELSE ASSEMBLE A);

(LC IS REF 0; LA IS REF 0; INST IS REF 0;

LC := LC1; ASSEMBLE(#() (LA := LA1; GOTO LC; LA1: VAL INST));

GOTO DONE;

LC1: COMPILE(# X (LC := LC2; INST := X; GOTO LA; LC2:));GOTO ERROR;

DONE:);

OS

GEDANKEN ()

PARSE IS # (IN, AMB, FAIL)

(U IS REF 0;

PC IS # () (C IS IN(); IF C = "." THEN GOTO FAIL ELSE C);

PS ISR # () (C1 IS PC(); AMB(L1); U := (C1, PS(), PC()); GOTO L2;

L1: U := C1; L2: VAL U);

(R IS PS(); IF IN() = "." THEN R ELSE GOTO FAIL));

PRNSQ ISR # X IF ISFUNCTION X THEN

(PRINT "("; VECTOR(1, X UL, #I PRNSQ X I); PRINT ")")

ELSE PRINT X;

PRNSQ

(C IS REF NIL; W IS REF NIL; R IS REF NIL; CHAR IS REF NIL;

C := (PARSE(#() (W := (L1, VAL W); GOTO CONT; L1: VAL CHAR),

L2 (R := (L2, VAL R)), CONT),

VAL C);

CONT: IF R = NIL AND W = NIL THEN GOTO DONE

ELSE IF R = NIL THEN (CHAR := READCHAR(); R := W; W := NIL)

ELSE ();

(L IS R 1; R := R 2; GOTO L);

DONE: VAL C);

OSA.

GEDANKEN ()

PARSE IS # (IN, AMB, FAIL)

(U IS REF 0;

PC IS # () (C IS IN(); IF C = "." THEN GOTO FAIL ELSE C);

PS ISR # () (C1 IS PC(); AMB(L1); U := (C1, PS(), PC()); GOTO L2;

L1: U := C1; L2: VAL U);

```

(R IS PS()); IF INC() = "." THEN R ELSE GOTO FAIL));
PRNSQ ISR # X IF ISFUNCTION X THEN
  (PRINT "("; VECTOR(1,X UL, #I PRNSQ X I); PRINT ")")
  ELSE PRINT X;
PRNSQ
(C IS REF NIL; W IS REF NIL; R IS REF NIL; CHAR IS REF NIL;
 C := (PARSE( #() (W := (L1, VAL W); GOTO CONT; L1: VAL CHAR),
 # L2 (R := (L2, VAL R)), CONT),
 VAL C);
CONT: IF R = NIL AND W = NIL THEN GOTO DONE
  ELSE IF R = NIL THEN (CHAR := READCHAR(); R := W; W := NIL)
  ELSE ();
(L IS R 1; R := R 2; GOTO L);
DONE: VAL C);
0$ABCDE.

```

```

GEDANKEN ()
PARSE IS # (IN, AMB, FAIL)
(U IS REF 0;
 PC IS # () (C IS INC(); IF C = "." THEN GOTO FAIL ELSE C);
 PS ISR # () (C1 IS PC(); AMB(L1); U := (C1, PS(), PC()); GOTO L2;
 L1: U := C1; L2: VAL U);
(R IS PS()); IF INC() = "." THEN R ELSE GOTO FAIL));
PRNSQ ISR # X IF ISFUNCTION X THEN
  (PRINT "("; VECTOR(1,X UL, #I PRNSQ X I); PRINT ")")
  ELSE PRINT X;

```

```

PRNSQ
(C IS REF NIL; W IS REF NIL; R IS REF NIL; CHAR IS REF NIL;
 C := (PARSE( #() (W := (L1, VAL W); GOTO CONT; L1: VAL CHAR),
 # L2 (R := (L2, VAL R)), CONT),
 VAL C);
CONT: IF R = NIL AND W = NIL THEN GOTO DONE
  ELSE IF R = NIL THEN (CHAR := READCHAR(); R := W; W := NIL)
  ELSE ();
(L IS R 1; R := R 2; GOTO L);
DONE: VAL C);
0$ABCD.

```

```

GEDANKEN ()
PRNV UNITSEQ 17;
PRINT (UNITSEQ 17) REF 1;
PRNV VECTOR(3,0,INC);
PRNV VECTOR(3,2,INC);
PRNV VECTOR(3,3,INC);
PRNV VECTOR(REF 3, REF 5, NCREF REF INC);
PRINT(VECTOR(3,0,INC))REF LL;
PRINT(VECTOR(3,5,INC))REF 4;
PRNV PVECTOR(0,9,INTTODIGIT);
PRNV PVECTOR(0,9, #I DIGITTOINT INTTODIGIT I);
0$

```

```

GEDANKEN ()
CONS IS #(X, Y) #Z IF Z = 1 THEN X ELSE Y;
CAR IS #X X 1;
CDR IS #X X 2;
X IS CONS(CONS(1,2),CONS(3,4));
PRNV (CAR CAR X, CDR CAR X, CAR CDR X, CDR CDR X);
0$

```

```

GEDANKEN ()
LISTLENGTH ISR #L IF L = NIL THEN 0 ELSE INC LISTLENGTH L 2;
LISTELEM ISR #(I, L) IF L = NIL THEN GOTO ERROR

```

```

ELSE IF I = 1 THEN L 1 ELSE LISTELEM(DEC I, L 2);
APPEND ISR #(X, Y) IF X = NIL THEN Y ELSE (X 1, APPEND(X 2, Y));
VECTOR ISR #(L, U, F)
IF GREATER(L, U) THEN
  # I IF I = LL THEN L ELSE IF I = UL THEN DEC L ELSE GOTO ERROR
ELSE (V IS VECTOR(L, DEC U, F); T IS F U;
  # I IF I = UL THEN U ELSE IF I = U THEN T ELSE V I);
(
MAKESEQFROMLIST IS # L
# I IF I = LL THEN 1 ELSE IF I = UL THEN LISTLENGTH L
ELSE LISTELEM(I, L);
MAKELISTFROMSEQ ISR # S MLFSI(1, S);
MLFSI ISR #(I, S) IF GREATER(I, S UL) THEN NIL
ELSE # K (CASE K OF S I, MLFSI(INC I, S));
(X IS (7, (8, (9, NIL))))); A IS APPEND(X, X);
PRINT LISTLENGTH NIL;
PRINT LISTLENGTH X;
PRNV PVECTOR(1, 3, #I LISTELEM(I, X));
PRNV PVECTOR(1, 6, #I LISTELEM(I, A));
PRNV MAKESEQFROMLIST NIL;
PRNV MAKESEQFROMLIST X;
PRINT MAKELISTFROMSEQ ();
PRNV PVECTOR(1, 3, #I LISTELEM(I, MAKELISTFROMSEQ (7, 8, 9)));
PRINT NIL));
OS

```

```

GEDANKEN ()
PROPVAL IS #(P, L)
(P IS COERCE P;
SEARCHL ISR # X
IF X = NIL THEN
  (NEWV IS REF 0; L := ((P, NEWV), VAL L); NEWV)
ELSE IF (X 1) 1 = P THEN (X 1) 2 ELSE SEARCHL X 2;
SEARCHL VAL L);
MAKEPROPLIST IS #() (L IS REF NIL; # P PROPVAL(P, L));
A IS MAKEPROPLIST(); B IS MAKEPROPLIST();
PROP IS ATOM();
PRINT VAL A PROP;
PRINT VAL A PROP;
A PROP := 100;
PRINT VAL A PROP;
A "Q" := 101; A 17 := 102; A PROP := 103;
B "Q" := 201; B 17 := 202; B PROP := 203;
PRNVR(A "Q", A 17, A PROP, A TRUE, B "Q", B 17, B PROP);
OS

```

```

GEDANKEN ()
R4 IS REF 4; M3 IS REF NEG 3;
NOT IS #X IF X THEN FALSE ELSE TRUE;
NEG ISR #X (X IS COERCE X; IF NOT ISINTEGER X THEN GOTO ERROR
ELSE IF X = 0 THEN 0 ELSE IF GREATER(X, 0) THEN DEC NEG DEC X
ELSE INC NEG INC X);
ADD ISR #(X, Y) (X IS COERCE X; Y IS COERCE Y;
IF NOT ISINTEGER X OR NOT ISINTEGER Y THEN GOTO ERROR
ELSE IF X = 0 THEN Y ELSE IF GREATER(X, 0) THEN INC ADD(DEC X, Y)
ELSE DEC ADD(INC X, Y));
SUBTRACT ISR #(X, Y) (X IS COERCE X; Y IS COERCE Y; ADD(X, NEG Y));
MULTIPLY ISR #(X, Y) (X IS COERCE X; Y IS COERCE Y;
IF NOT ISINTEGER X OR NOT ISINTEGER Y THEN GOTO ERROR
ELSE IF X = 0 THEN 0
ELSE IF GREATER(X, 0) THEN ADD(MULTIPLY(DEC X, Y), Y)
ELSE SUBTRACT(MULTIPLY(INC X, Y), Y));

```

```

DIVIDE ISR #(X, Y) (X IS COERCE X; Y IS COERCE Y;
  IF NOT ISINTEGER X OR NOT ISINTEGER Y OR Y = 0 THEN GOTO ERROR
  ELSE IF GREATER(0, Y) THEN NEG(DIVIDE(X, NEG Y))
  ELSE IF NOT GREATER(Y, X) THEN INC DIVIDE(SUBTRACT(X, Y), Y)
  ELSE IF NOT GREATER(Y, NEG X)
    THEN DEC DIVIDE(ADD(X, Y), Y) ELSE 0);
REMAINDER ISR #(X, Y) (X IS COERCE X; Y IS COERCE Y;
  SUBTRACT(X, MULTIPLY(Y, DIVIDE(X, Y))));
PRINT NOT TRUE;
PRINT NOT REF FALSE;
PRINT NEG 0;
PRINT NEG 3;
PRINT NEG M3;
PRINT ADD(0, 4);
PRINT ADD(1, R4);
PRINT ADD(M3, 4);
PRINT SUBTRACT(M3, R4);
PRINT MULTIPLY(0, 4);
PRINT MULTIPLY(2, R4);
PRINT MULTIPLY(M3, 4);
PRINT DIVIDE(R4, 3);
PRINT DIVIDE(2, 3);
PRINT DIVIDE(0, 3);
PRINT DIVIDE(NEG 2, 3);
PRINT DIVIDE(NEG 4, 3);
PRINT DIVIDE(NEG 4, M3);
PRINT REMAINDER(R4, 3);
PRINT REMAINDER(NEG 4, M3);
OS

```

GEDANKEN ()

```

LAB: (A1 IS ATOM(); A2 IS ATOM(); F IS REF 0;
  L IS REF (REF 1, TRUE, FALSE, "A", LL, UL, IMPREF(NEG, ATOM), NEG, LAB,
  ERROR);
  ITER ISR #I IF GREATER(I, L UL) THEN ()
  ELSE (PRINT F L I; ITER INC I);
  F := ISINTEGER ; ITER 1;
  F := ISBOOLEAN ; ITER 1;
  F := ISCHAR ; ITER 1;
  F := ISATOM ; ITER 1;
  F := ISFUNCTION ; ITER 1;
  F := ISREF ; ITER 1;
  F := ISLABEL ; ITER 1;
  L := ((1, 1), (1, 2),
  (TRUE, REF TRUE), (TRUE, REF FALSE), (FALSE, REF FALSE),
  (REF "A", REF "A"), (REF "A", REF "B"),
  (REF LL, LL), (REF LL, UL), (REF UL, UL),
  (A1, A1), (A1, A2),
  (NEG, NEG), (LAB, LAB),
  (1, TRUE), (TRUE, "A"), ("A", LL), (LL, A1), (A1, NEG),
  (NEG, LAB), (LAB, 1));
  F := EQUAL; ITER 1);

```

OS
/*