

```

EEEEEEEEEE
EEEEEEEEEE
EE
EE
EE
EE
EEEEEEEE
EEEEEEEE
EE
EE
EE
EE
EEEEEEEEEE
EEEEEEEEEE

```

```

....
....
....
....

```

Alan

```

44 44 000000 000000 44 44 11 44 44
44 44 000000 000000 44 44 11 44 44
44 44 00 00 00 00 00 44 44 1111 44 44
44 44 00 00 00 00 00 44 44 1111 44 44
44 44 00 0000 00 0000 44 44 11 44 44
44 44 00 0000 00 0000 44 44 11 44 44
4444444444 00 00 00 00 00 00 4444444444 11 4444444444
4444444444 00 00 00 00 00 00 4444444444 11 4444444444
44 0000 00 0000 00 44 11 44
44 0000 00 0000 00 44 11 44
44 00 00 00 00 44 11 44
44 00 00 00 00 44 11 44
44 000000 000000 44 111111 44
44 000000 000000 44 111111 44

```

```

W W EEEEE L H H AAA M M BBBB
W W E L H H A A MM MM B B
W W E L H H A A M M B B
W W EEEE L HHHH A A M M BBBB
W W W E L H H AAAAA M M B B
WW WW E L H H A A M M B B
W W EEEEE LLLL H H A A M M BBBB

```

```

LPTSPL VERSION 6(344) RUNNING ON LPT500
*START* USER WELHAM B [400,414] JOB E SEQ. 3237 DATE 17-NOV-75 15:06:13 MONITOR NETMON 5.07B V3 (AUG 12 T *START*
REQUEST CREATED: 17-NOV-75 15:07:17
FILE: DSKA1:E[400,414] CREATED: 17-NOV-75 15:02:00 <155> PRINTED: 17-NOV-75 15:06:21
QUEUE SWITCHES: /PRINT:ARROW /FILE:ASCII /COPIES:2 /SPACING:1 /LIMIT:38 /FORMS:NORMAL
FILE WILL BE RENAMED TO <055> PROTECTION

```

* PROLOG EQUATION SOLVING PROGRAM.
* BOB WELHAM 1975.

*****.
* SOLVE ONE EQUATION IN ONE UNKNOWN.
*****.

+SOLVE11(*EQN,*U,*ANS)-SORTER(SOLVING-*EQN-FOR-*U)-LIGNE-LIGNE
-SOL11(*EQN,*U,*ANS)-ANSWER(*ANS).

* LOOK FOR SPECIAL CASES FIRST.

+SOL11(*LHS=0,*U,*U=*SOL)-LINEAR(*U,*LHS,*A,*B)-/
-SIMPLIFY(-1.*B.(*A:-1),*SOL).

+SOL11(*LHS=0,*U,*U=*SOL1 # *U=*SOL2)-QUAD(*U,*LHS,*A,*B,*C)-/
-TRACE(QUADRATIC-EQUATION-IN-*U)-TRACE(COEFFICIENTS-*A-*B-*C)
-SIMPLIFY((*B:2+-1.4.*A.*C):(2:-1),*SQRT)
-SIMPLIFY((-1.*B+*SQRT).(2.*A):-1,*SOL1)
-SIMPLIFY((-1.*B-1.*SQRT).(2.*A):-1,*SOL2).

* THEN TRY BASIC METHOD OF ISOLATION, COLLECTION AND ATTRACTION.

+SOL11(*EQN,*U,*ANS)-ISOLATE(*U,*EQN,*ANS)-/.

+SOL11(*EQN,*U,*ANS)-COLLECT(*U,*EQN,*NEW)-SOL11(*NEW,*U,*ANS).

+SOL11(*EQN,*U,*ANS)-ATTRACT(*U,*EQN,*NEW)-SOL11(*NEW,*U,*ANS).

*****.
* ISOLATION ROUTINES.
*****.

+ISOLATE(*X,*EQN1#*EQN2,*ANS1#*ANS2)-/
-ISOLATE(*X,*EQN1,*ANS1)-ISOLATE(*X,*EQN2,*ANS2).

+ISOLATE(*X,*X=*RHS,*X=*SOL)-FREEOF(*RHS,*X)-/SIMPLIFY(*RHS,*SOL).

+ISOLATE(*X,*LHS=*RHS,*ANS)
-PERM2(*LHS,*RHS,*L,*R)-FREEOF(*R,*X)-SINGLEOCC(*L,*X)
-TRACE(ISOLATION-ON-*LHS=*RHS-FOR-*X)
-SETUP(*X,*L,*LM)-ISOLATES(*AXIOM)
-APPLY(LTR,*AXIOM,*LM=*R,*NEW)
-ISOLATE(*X,*NEW,*ANS).

+SETUP(*X,*U,*V,*P,*Q)-/--PERM2(*U,*V,*P,*Q)-SINGLEOCC(*P,*X).
+SETUP(*X,*U,*V,*P,*Q)-/--PERM2(*U,*V,*P,*Q)-SINGLEOCC(*P,*X).
+SETUP(*X,*E,*E).

*****.
* COLLECTION ROUTINES.
*****.

+COLLECT(*X,*OLD,*NEW)-OCC(*OLD,*X,*N)-INF(*N,2)-/--FAIL.

+COLLECT(*X,*OLD,*NEW)-TRACE(COLLECTING-*X-IN-*OLD)
-OCC(*OLD,*X,*N)
-COLLECTS(*AXIOM:*HOW)-APPLY(*HOW,*AXIOM,*OLD,*NEW)
-OCC(*NEW,*X,*M)-INF(*M,*N)
-TRACE(COLLECTION-SUCCESSFUL).

* TRY TO COLLECT WITHIN A SUB-TERM.

+COLLECT(*X,*OLD,*NEW)-UNIV(*OLD,*F,*A.NIL)
-COLLECT(*X,*A,*B)-UNIV(*NEW,*F,*B.NIL).

+COLLECT(*X,*OLD,*NEW)-UNIV(*OLD,*OP,*ARG1,*ARG2.NIL)
-COLLECT(*X,*ARG1,*ARG3)-UNIV(*NEW,*OP,*ARG3,*ARG2.NIL).

+COLLECT(*X,*OLD,*NEW)-UNIV(*OLD,*OP,*ARG1,*ARG2.NIL)
-COLLECT(*X,*ARG2,*ARG3)-UNIV(*NEW,*OP,*ARG1,*ARG3.NIL).

*****.
* ATTRACTION ROUTINES.
*****.

+ATTRACT(*X,*OLD,*NEW)-OCC(*OLD,*X,*N)-INF(*N,2)-/--FAIL.

+ATTRACT(*X,*OLD,*NEW)-TRACE(ATTRACTING-*X-IN-*OLD)
-ATTRACTS(*AXIOM:*HOW)-APPLY(*HOW,*AXIOM,*OLD,*NEW)
-TRACE(ATTRACTION-SUCCESSFUL).

* TRY TO ATTRACT WITHIN SUB-TERM.

```

+ATTRACT(*X,*OLD,*NEW)-UNIV(*OLD,*F,*A.NIL)
-ATTRACT(*X,*A,*B)-UNIV(*NEW,*F,*B.NIL).

+ATTRACT(*X,*OLD,*NEW)-UNIV(*OLD,*OP,*ARG1,*ARG2.NIL)
-ATTRACT(*X,*ARG1,*ARG3)-UNIV(*NEW,*OP,*ARG3,*ARG2.NIL).

+ATTRACT(*X,*OLD,*NEW)-UNIV(*OLD,*OP,*ARG1,*ARG2.NIL)
-ATTRACT(*X,*ARG2,*ARG3)-UNIV(*NEW,*OP,*ARG1,*ARG3.NIL).

```

```

*****
* SIMPLIFICATION ROUTINES.
*****

```

* QUICK KILLS.

```

+SIMPLIFY(*E,*E)-UNIV(*E,*CONSTANT.NIL)-/.
+SIMPLIFY(-1,-1)-/.
+SIMPLIFY(*E,*E)-WONTSIMP(*E)-/.

```

```

+SIMPLIFY(*E,*R)-TRACE(SIMPLIFYING-*E)
-SIMPLIFES(*AX)
-APPLY(LTR,*AX,*E,*S)-/-SIMPLIFY(*S,*R).

```

```

+SIMPLIFY(*E,*R)-EVAL(*E,*R)-/-TRACE(*E-EVALUATED-TO-*R).

```

```

+SIMPLIFY(*E,*R)-UNIV(*E,*OP,*E1,*E2.NIL)
-SIMPLIFY(*E1,*R1)-SIMPLIFY(*E2,*R2)-DIFF(*E1,*E2,*R1,*R2)-/
-UNIV(*F,*OP,*R1,*R2.NIL)-SIMPLIFY(*F,*R).

```

```

+SIMPLIFY(*E,*R)-UNIV(*E,*F,*A.NIL)
-SIMPLIFY(*A,*B)-DIFF(*A,*B)-/
-UNIV(*D,*F,*B.NIL)-SIMPLIFY(*D,*R).

```

```

+SIMPLIFY(*E,*E)-AJOUT((+WONTSIMP(*E)).NIL).

```

*+Simplify (*E, *R)*
*- UNIV (*E, *F, *ARGS)*
*- MAPLIST (*ARGS, SIMPLIFY, *ARGS1)*
*- APPLY (*ARGS, *ARGS1, DIFF) - /*
*- UNIV (*E1, *F, *ARGS1) - SIMPLIFY (*E1, *R) .*

```

*****
* DESCRIPTION ROUTINES.
*****

```

```

+LINEAR(*U,*E,0,*E)-FREEOF(*E,*U)-/.

```

```

+LINEAR(*U,*U,1,0)-/.

```

```

+LINEAR(*U,*E1+*E2,*A,*B)
-LINEAR(*U,*E1,*A1,*B1)-LINEAR(*U,*E2,*A2,*B2)-/
-SIMPLIFY(*A1+*A2,*A)-SIMPLIFY(*B1+*B2,*B).

+LINEAR(*U,*E1.*E2,*A,*B)-PERM2(*E1,*E2,*E3,*E4)
-FREEOF(*E3,*U)-LINEAR(*U,*E4,*A4,*B4)
-SIMPLIFY(*E3.*A4,*A)-SIMPLIFY(*E3,*B4,*B).

+QUAD(*U,*E,0,*B,*C)-LINEAR(*U,*E,*B,*C)-/.
+QUAD(*U,*U.*U,1,0,0)-/.
+QUAD(*U,*U:2,1,0,0)-/.

+QUAD(*U,*E1+*E2,*A,*B,*C)
-QUAD(*U,*E1,*A1,*B1,*C1)-QUAD(*U,*E2,*A2,*B2,*C2)-/
-SIMPLIFY(*A1+*A2,*A)
-SIMPLIFY(*B1+*B2,*B)
-SIMPLIFY(*C1+*C2,*C).

+QUAD(*U,*E1.*E2,*A,*B,*C)
-PERM2(*E1,*E2,*E3,*E4)-FREEOF(*E3,*U)-/
-QUAD(*U,*E4,*A4,*B4,*C4)
-SIMPLIFY(*E3.*A4,*A)
-SIMPLIFY(*E3.*B4,*B)
-SIMPLIFY(*E3.*C4,*C).

+QUAD(*U,*E1.*E2,*A,*B,*C)
-LINEAR(*U,*E1,*A1,*B1)-LINEAR(*U,*E2,*A2,*B2)
-SIMPLIFY(*A1.*A2,*A)
-SIMPLIFY(*A1.*B2+*A2.*B1,*B)
-SIMPLIFY(*B1.*B2,*C).

```

```

*****
* SIMULTANEOUS EQUATIONS ROUTINES.
*****

```

```

+SIMSOLVE(TRUE,NIL,TRUE)-/.

+SIMSOLVE(*ES1#*ES2,*US,*ANS1#*ANS2)
-TRACE(DISJUNCTION-*ES1-OR-*ES2)
-SIMSOLVE(*US,*ANS1,*ES1)-SIMSOLVE(*ES2,*US,*ANS2).
ES1, US, ANS1

+SIMSOLVE(*ES,*US,*ANS&*AR)
-TRACE(SIMULTANEOUSLY-SOLVING-*ES-FOR-*US)
-SELECT(*U,*US,*UR)-SELECTA(*EQN,*ES,*ER)
-CONTAINS(*EQN,*U)-SOLVE11(*EQN,*U,*ANS)-/
-SUBSTITUTE(*ANS,*ER,*NEWEQNS)

```

-SIMSOLVE(*NEWQNS,*UR,*AR).

+FIN.

• COP PROOS.DAT = initial state

• COP TEXT.DAT = EQNSI

• RU PSO (prolog)

-T, (trace on)

-TTY,

}

-T-TTY,

~