

PHOTO

12/18/81

FILE = TST

BLK= 0

01( TEST SHIT TO DUMP NEAT OVECTOR STUFF )

11HEX V= VERBADR

21SUBR NEWINTR B LDAX, B INX, A L MOV, B LDAX, B INX, A H MOV,

31VERBADR SHLD, PCHL,

41CODE ZAMMER NEWINTR Y LXIX, NEXT

51: LSAT ZAMMER ;

61DECIMAL ;S

71: VD CR ." C= " DUP NOWC OVBe .

81." R= " DUP NOWR OVBe .

91." D= " DUP NOWD OVBe .

101." DIS= " DUP DISTANCE OV@ H.

111." DD= " DUP DELTADIST OV@ H.

121." MAXD= " DUP MAXDIST OVBe H.

131." X= " DUP VX OV@ H.

141." Y= " VY OV@ H. CR ;

151-->

FILE = VE

BLK= 0

0!( VECTOR FIELDS ) ." \*\*OGGBUG" ( VLENGTH 8 - C= VLENGTH )  
1!DECIMAL VLENGTH SC= INTR NC= INTC ( INITIAL POS AND COL )  
2!NC= NOWR NC= NOWC ( CURRENT ROW AND COLUMN )  
3!NC= NOWD ( CURRENT DIRECTION )  
4!( NC= NXTR NC= NXTC ) ( NEXT ROW AND COLUMN )  
5!( CUSTOM VECTOR ROUTINE GOODIES )  
6!NC= BASEX 1+  
7!NC= BASEY 1+  
8!NC= DELTAX 1+  
9!NC= DELTAY 1+  
10!NC= MAXDIST MAXDIST NOWR - 1+ C= POSLEN  
11!NC= DISTANCE 1+  
12!NC= DELTADIST 1+ DELTADIST NOWR - C= SNATLEN  
13!( NC= ACCDIST 1+ )  
14!NC= MEMDIST NC= MEMR NC= MEMC NC= MEMD -->  
15!-->

BLK= 1

0!( MORE CUSTOM VECTOR FIELDS )  
1!NC= CUSVEC 1+ ( CUSTOM VECTOR ROUTINE ADDRESS )  
2!NC= MYTYPE ( VECTOR TYPE INDICATOR )  
3!NC= MYFLAG ( BUILD IN NEATO FLAG ) NC= FLAGCODE  
4!NC= MYFACE 1+ ( WHAT I LOOK LIKE IN THE OPEN )  
5!NC= VCOR 1+ ( MY COROUTINE CELL )  
6!NC= BEHIND 1+ ( FELLOW BEHIND ME )  
7!NC= AHEAD 1+ ( FELLOW AHEAD OF ME )  
8!NC= VISFLAG  
9!1+ C= MLENGTH  
10!MLENGTH SC= HOSSV NC= ASSMSV  
11!NC= VIRGIN  
12!NC= DIST-1 ( PREV DISTANCE )  
13!NC= DISPF ( DISPLACEMENT FACTOR )  
14!NC= SNATCHER 1+  
15!1+ C= HLENGTH -->

BLK= 2

0!( MORE UNIQUE VECTOR STUFF )  
1!MLENGTH AHEAD C= MYSLAVE  
2!SC= FNDPTR FNDPTR C= TRACKPTR 1+ NC= TREECK 1+  
3!BEHIND C= MYBOSS  
4!NC= FRONTIER 1+  
5!NC= VISMAT NCOLS + C= TREES  
6!0 SC= TPL NC= TPH NC= TC NC= TR NC= TD 1+ C= TEL  
7!TEL NNODES \* C= TDEPTH 70 C= SURPLUS  
8!TDEPTH TREES + SURPLUS +  
9!1+ C= MONLEN  
10!HLENGTH C= PLENGTH ( PLAYERS VECTOR LENGTH )  
11!PLENGTH C= RVLENGTH ( REVEALERS LENGTH )  
12!( BITS AND CODES )  
13!( VECTOR TYPES )  
14!0 SC= T-TYP NC= H-TYP NC= M-TYP C= K-TYP  
15!-->

FILE = VE

BLK= 3

```
0:( HOSTAGE AND PLAYER STATE VARIABLES )
1:( THE HOSTAGE STATE VARIABLE )
2:0 SC= HSFREE ( HOSTAGE FREE )
3:INC= HSATP ( HOSTAGE ATTACHED TO PLAYER )
4:INC= HSATM DROP ( HOSTAGE ATTACHED TO MONSTER )
5:( ASSIMILATION STATE VARIABLE )
6:0 SC= ASNOT ( NOT ASSIMILATED )
7:INC= ASSIM DROP ( FULLY ASSIMILATED )
8:( PLAYERS ASSIMILATION STATE VARIABLE )
9:0 SC= ASCOOL ( PLAYER IS SPIFFY )
10:INC= ASONTOP DROP ( PLAYER IS ON TOP OF HOSTAGES )
11:-->
12:
13:
14:
15:
```

BLK= 4

```
0:( VECTOR STUFF )
1:XC? NOT IFTRUE VPTR @ HEX FFF0 VPTR ! IFEND <STKD
2:RAMMARK MLENGTH BR= BKGV RAMLEN C= BKVL VARHERE C= BKVS
3:RAMMARK PLENGTH BR= PLYRV RAMLEN C= PLYRL VARHERE C= PLYVS
4:RAMMARK RVLENGTH BR= REVV RAMLEN C= REVL VARHERE C= REVS
5:RAMMARK MLENGTH BR= TV1 RAMLEN C= TVVL VARHERE C= TVVS
6:-->
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

BLK= 5

```
0:( MONSTER STUFF )
1:
2:RAMMARK MONLEN BR= MONV1
3:MONLEN BR= MONV2
4:MONLEN BR= MONV3
5:MONLEN BR= MONV4
6:RAMLEN C= MONVL VARHERE C= MONVS
7:MONLEN C= MONVBYTES
8:STK> XC? NOT IFTRUE VPTR @ H. VPTR ! IFEND
9:DECIMAL -->
10:
11:
12:
13:
14:
15:
```

FILE = VE

BLK= 6

```
01( TREASURE VECTORS )
11
21RAMMARK MLENGTH BR= TRSV1
31MLENGTH BR= TRSV2 MLENGTH BR= TRSV3
41MLENGTH BR= TRSV4
51RAMLEN C= TRSVL VARHERE C= TRSVS
61MLENGTH C= TRSVBYTES
714 C= TOTAL-JEWELS
81-->
91
101
111
121
131
141
151
```

BLK= 7

```
01( HOSTAGE VECTORS )
11RAMMARK HLENGTH BR= HOSV1
21HLENGTH BR= HOSV2 HLENGTH BR= HOSV3
31HLENGTH BR= HOSV4
41RAMLEN C= HOSVL VARHERE C= HOSVS
51HLENGTH C= HOSVBYTES
614 C= TOTAL-HOSTAGES
71TABLE HOSTAB HOSV1 , HOSV2 , HOSV3 , HOSV4 , 0 ,
81
91( ***** )
101HOSV4 C= TEMRM
111-->
121
131
141
151
```

BLK= 8

```
01( MORE NEAT VECTOR STUFF )
11: ZAP:VECT 0 MONV4 BKGV MONV4 - BKVL + FILL
210 HOSV4 TRSV1 HOSV4 - MLENGTH + FILL ;
31-->
41
51
61
71
81
91
101
111
121
131
141
151
```

FILE = VE

BLK= 9

0: ( SPECIAL VECTOR GETTERS AND PUTTERS )

1: CODE PUSH:CCR O H MVI, H D MOV, Y PUSHX, vaddr LIYD,

2: NOWC Y L LDX, NOWR Y E LDX, Y POPX, H PUSH, D PUSH, NEXT

3:

4: CODE PUSH:CCRD O H MVI, H D MOV, Y PUSHX, vaddr LIYD,

5: NOWC Y L LDX, NOWR Y E LDX, NOWD Y A LDX,

6: Y POPX, H PUSH, D PUSH, A E MOV, D PUSH, NEXT

7:

8: CODE COGO ( exchange BC with VCOR )

9: vaddr LHLD, VCOR D LXI, D DAD,

10: M A MOV, C M MOV, A C MOV, H INX,

11: M A MOV, B M MOV, A B MOV, NEXT

12: SETCO 1+ VCOR V! ;

13:

14:

15: -->

FILE = VA

BLK= 0

```
01( GAME CONTROL PARAMETERS )
11BV= NOBREAK
21V= TRASHFLAG
31V= GAME-OVER V= GAME#
41V= NPLAYERS V= PLAYERUP
51V= INITIAL-LIVES
61V= REMAINING-LIVES
71V= PLAYERDEAD ( PLAYER NAILED BY MONSTER FLAG )
81V= PLAYERVELO ( PLAYER VELOCITY )
91BV= FREEZEFLAG ( PLAYER MOTION FREEZE FLAG )
101BV= SMARTS ( MONSTER SMARTNESS FACTOR )
111V= MONSTERCOUNT ( # OF MONSTERS FLOATING AROUND )
121BV= BANC BV= BANR ( POINT OF BANISHMENT FOR MONSTER )
131BV= IBNC BV= IBNR ( POINT OF INITIAL RETURN FOR MONSTER )
141-->
151
```

BLK= 1

```
01( MORE VARIABLES )
11V= TOTAL-CONNECTS V= OLD-CONNECTS
21V= TOTAL-REVEALED-GROTTO
31V= KEY-THRESHOLD
41BV= KEY-STATUS
510 SC= KYNONE NC= KYSHOW NC= KYOPEN NC= KYGONE DROP
61V= TOTAL-PATHS
71V= REVEALED-PATHS ( # OF PATHS REVEALED TO PLAYER SO FAR )
81BV= REVEAL-ACTIVE
91V= START-COL V= STOP-COL
101V= FOUNDIT BV= THATSALL
111DECIMAL -->
121
131
141
151
```

BLK= 2

```
01( FREEZE AND UNFREEZE ROUTINES )
11SUBR FREEZE FREEZEFLAG H LXI, M INR, RET,
21SUBR FREEZE? FREEZEFLAG LDA, A ANA, RET,
31CODE FREEZETH FREEZE CALL, NEXT
41CODE UNFREEZE FREEZEFLAG H LXI, M DCR, 0<, IF, 0 M MVI, THEN,
51NEXT
61-->
71
81
91
101
111
121
131
141
151
```

FILE = DI

BLK= 0

```
0:( NEW SQUARE ROOT ROUTINE )
1:IF= sqrt1
2:SUBR sqrt <ASSEMBLE
3:1 A MVI, 1 B LXI, 1 D LXI,
4:LABEL sqrt1 A ANA, D DSBC, RZ, RC, D DAD, B INX, B INX,
5:XCHG, B DAD, A INR, XCHG, sqrt1 JMPR, ASSEMBLE>
6:-->
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

BLK= 1

```
0:( 16 BIT INTEGER DIVIDE ROUTINE: M N UN/ Q R ) DECIMAL
1:FORWARD .ZERO FORWARD IDV50 FORWARD IDV60
2:FORWARD IDV10 FORWARD IDV20 FORWARD IDV30 FORWARD IDV40
3:SUBR unsddiv <ASSEMBLE L C MOV, H B MOV, D A MOV, O H LXI,
4:E ORA, .ZERO JRZ, B A MOV, 16 B MVI,
5:LABEL IDV10 C RALR, RAL, H DADC, D DSBC,
6:LABEL IDV20 CMC, IDV50 JRNC,
7:LABEL IDV30 IDV10 DJNZ, IDV60 JMPR,
8:LABEL IDV40 C RALR, RAL, H DADC, A ANA, D DADC,
9:IDV30 JRC, IDV20 JRZ,
10:LABEL IDV50 IDV40 DJNZ, D DAD, A ANA, ( MAKE IT POS )
11:LABEL IDV60 C RALR, RAL, A D MOV, C E MOV,
12:LABEL .ZERO RET, ASSEMBLE>
13:SUBR UNSDIV H PUSH, D DSBC, CY, IF, O D LXI, H POP, ELSE,
14:H POP, unsddiv CALL, THEN, RET, CODE UN/ EXX, D POP, H POP,
15:UNSDIV CALL, H PUSH, D PUSH, EXX, NEXT DECIMAL -->
```

BLK= 2

```
0:( COMPUTE DELTA FOR 1 COORDINATE - CLEAR VECTOR )
1:( FIRST A NEGATION SUBROUTINE )
2:SUBR CMPHL H A MOV, CMA, A H MOV, L A MOV, CMA, A L MOV, H INX,
3:RET,
4:( IN: HL=TARGET, DE=TIME, BC=START )
5:SUBR CDELTA B PUSH, A ANA, B DSBC, CY~, IF, UNSDIV CALL,
6:ELSE, CMPHL CALL, UNSDIV CALL, CMPHL CALL, XCHG, CMPHL CALL,
7:XCHG, THEN, B POP, B DAD, RET,
8:DECIMAL -->
9:
10:
11:
12:
13:
14:
15:
```



FILE = NM

BLK= 0

```
01( MESH PARAMETERS ) <STKD
11
21336 NCOLS / C= COLSIZE 180 NROWS / C= ROWSIZE
3140 C= COLGUARD 28 C= ROWGUARD 10 C= CIR-RAD
418 C= HOLE-RAD NROWS 1- C= START-ROW
51COLSIZE COLGUARD - C= COLDEV ROWSIZE ROWGUARD - C= ROWDEV
61
71: COLCENT COLSIZE * COLSIZE 2 / + 168 - ;
81: ROWCENT ROWSIZE * ROWSIZE 2 / + 107 - ;
91
101: COMP:X COLCENT COLDEV 2 / COLDEV RND - + ;
111: COMP:Y ROWCENT ROWDEV 2 / COLDEV RND - + ;
121
131: COMP:XY COMP:Y SWAP COMP:X SWAP ;
141STK> -->
151
```

BLK= 1

```
01( MESH MATRIX GOODIES )
110 SC= NODX NC= NODXH
21NC= NODY NC= NODYH NC= NBX 1+ NC= NBY 1+
31NC= MPLO 7 +
41NC= NDXO 7 +
51NC= NDYO 7 +
61NC= CONFLG NC= #CON
71NC= DRAWFLG NC= DRAWMSK
81NC= >TREASURE 1+
911+ C= NODSIZ
101NODSIZ NNODES * C= NODEMAT:SIZE
111NODEMAT:SIZE BA= NODEMAT -->
121
131
141
151
```

BLK= 2

```
01( NODE ZAMMERS )
11( SUBR node^ D= ROW E= COL C= DISP, OUT HL= ^ )
21F= N^1 F= N^2 SUBR node^ <ASSEMBLE D PUSH, B PUSH,
31D B MOV, B INR, NCOLS MINUS A MVI,
41LABEL N^1 NCOLS ADI, N^1 DJNZ, E ADD, A INR, A B MOV,
51NODSIZ MINUS H LXI, NODSIZ D LXI,
61LABEL N^2 D DAD, N^2 DJNZ, B DAD, O NODEMAT B LXI, B DAD,
71B POP, D POP, RET, ASSEMBLE>
81CODE NODE^ EXX, B POP, H POP, D POP, L D MOV, node^ CALL,
91H PUSH, EXX, NEXT
101SUBR noded^ node^ CALL, D PUSH, MPLO D LXI, D DAD, D POP, RET,
111-->
121
131
141
151
```

FILE = NM

BLK= 3

```
01( TEST:REL AND MOVE:NODE )
11( D=ROW,E=COL,C=REL COL ROW REL TEST:REL --- DIST )
21SUBR test:rel C A MOV, MPLO ADI, A C MOV, node^ CALL,
31M A MOV, RET,
41CODE TEST:REL EXX, B POP, H POP, D POP, L D MOV, test:rel CALL,
51A L MOV, O H MVI, H PUSH, EXX, NEXT
61( MOVE:NODE TABLES )
71DATA xtbl -1 B, 0 B, 1 B, -1 B, 1 B, -1 B, 0 B, 1 B,
81DATA ytbl 1 B, 1 B, 1 B, 0 B, 0 B, -1 B, -1 B, -1 B,
91SUBR move:node B PUSH, ( C=DIR, D=ROW,E=COL )
101O B MVI, ytbl H LXI, B DAD, M A MOV, D ADD, A D MOV,
111xtbl H LXI, B DAD, M A MOV, E ADD, A E MOV, B POP, RET,
121CODE MOVE:NODE EXX, B POP, H POP, D POP, L D MOV,
131move:node CALL, D L MOV, O D MVI, D H MOV,
141D PUSH, H PUSH, EXX, NEXT
151-->
```

BLK= 4

```
01( STUFF )
11: NODE! NODE^ ! ;
21: NODE@ NODE^ @ ;
31: NODEB@ NODE^ B@ ;
41: CLEAR:NODEMAT O O NODEMAT NODEMAT:SIZE FILL ;
51-->
61
71
81
91
101
111
121
131
141
151
```

BLK= 5

```
01( ESTVALDIR )
11F= EVDL
21SUBR estvaldir <ASSEMBLE NOWR Y D LDX, NOWC Y E LDX,
31O NOWD Y MVIX,
41LABEL EVDL NOWD Y A LDX, MPLO ADI, A C MOV, node^ CALL,
51M A MOV, A ANA, RNZ, NOWD Y INRX, EVDL JMPR, ASSEMBLE>
61CODE ESTVALDIR B PUSH, Y PUSHX, vaddr LIYD, estvaldir CALL,
71Y POPX, B POP, NEXT
81
91-->
101
111
121
131
141
151
```

FILE = NM

BLK= 6

01( NODE MATRIX MANIPULATORS )

11: SET:DRAWN ROLL DRAWMSK NODE^ SET ;

21: TEST:DRAWN ROLL DRAWMSK NODE^ BIT ;

31: SET:GROTTO:DRAWN DRAWFLG NODE^ BONE ;

41: TEST:GROTTO:DRAWN DRAWFLG NODEB@ ;

51-->

61

71

81

91

101

111

121

131

141

151

FILE = VC

BLK= 0

01( MORE STUFF )

11: RETURN:INITIAL:POSITION INTR VB@ NOWR VB! INTC VB@ NOWC VB! ;

21: SET:NEW:MCCR NOWR VB! NOWC VB! ;

31: SET:INITIAL:MCCR DUP ROLL INTR OVB! INTC OVB! ;

51: ~~BM\$HARGED\$FUNCTION~~ INTR VB@ = SWAP INTC VB@ = AND ;

61-->

71

81

91

101

111

121

131

141

151

FILE = CD

BLK= 0

```
01( COMPUTE DELTAS FOR STORAGE ROUTINE )
11( THIS ROUTINE COMPUTES DELTA FOR ONE COORDINATE )
21SUBR CDEL1 ( DE=R,C B=COORD PTR, C=DIR )
31B PUSH, D PUSH,
41B PUSH, C A MOV, MPLO ADI, A C MOV, node^ CALL, M L MOV,
51O H MVI, B POP, L A MOV, A ANA, O<>, IF,
61H PUSH, D PUSH, move:node CALL,
71B C MOV, node^ CALL, M E MOV, H INX, M D MOV, XCHG,
81XTHL, XCHG, node^ CALL, M C MOV, H INX, M B MOV,
91H POP, ( TARGET ) D POP, ( TIME ) CDELTA CALL, E A MOV,
101THEN, D POP, B POP, A B MOV, RET,
111-->
121
131
141
151
```

BLK= 1

```
01( SET DELTAS FOR BOTH COORDINATES FOR A GIVEN PATH )
11SUBR SETDELTS
21NBX B MVI, CDEL1 CALL, B PUSH, C A MOV, NDXO ADI, A C MOV,
31node^ CALL, B M MOV, B POP, NBY B MVI, CDEL1 CALL,
41B PUSH, C A MOV, NDYO ADI, A C MOV,
51node^ CALL, B M MOV, B POP, RET,
61-->
71
81
91
101
111
121
131
141
151
```

BLK= 2

```
01( COMPUTE DELTAS FOR WHOLE MATRIX )
11F= MAKELP
21CODE MAKEDELTS <ASSEMBLE B PUSH,
31O D LXI, O C MVI,
41LABEL MAKELP SETDELTS CALL,
51C A MOV, A INR, A C MOV, S CPI, MAKELP JRNZ, O C MVI,
61E A MOV, A INR, A E MOV, NCOLS CPI, MAKELP JRNZ, O E MVI,
71D A MOV, A INR, A D MOV, NROWS CPI, MAKELP JRNZ,
81B POP, NEXT ASSEMBLE>
91: FIXVGER NCOLS O DO NROWS O DO
101J I NODX NODE@ XADJ J I NBX NODE!
111J I NODY NODE@ YADJ J I NBY NODE! LOOP LOOP ;
121: MD FIXVGER MAKEDELTS ;
131-->
141
151
```

FILE = VR

BLK= 0

```
01( HOPPED UP 8 BIT MPY ROUTINE )
11( THIS ROUTINE IS USED TO MULTIPLY DELTA BY DISTANCE )
21( ADDING RESULT TO INITIAL DISP )
31( HL= INITIAL DISP, DE= DELTA, A= DIST )
41SUBR HOTMPY RRC, CY, IF, D DAD, THEN, E SLAR, D RALR,
51RRC, CY, IF, D DAD, THEN, E SLAR, D RALR,
61RRC, CY, IF, D DAD, THEN, E SLAR, D RALR,
71RRC, CY, IF, D DAD, THEN, E SLAR, D RALR,
81RRC, CY, IF, D DAD, THEN, E SLAR, D RALR,
91RRC, CY, IF, D DAD, THEN, E SLAR, D RALR,
101RRC, CY, IF, D DAD, THEN, E SLAR, D RALR,
111RRC, CY, IF, D DAD, THEN, RET,
121SUBR SQUARE BABS CALL, A E MOV, O D MVI, O H LXI,
131HOTMPY JMPR,
141--->
151
```

BLK= 1

```
01( CALCULATE X Y POSITION OF OBJECT FROM DISTANCE, BASE, AND )
11( DELTAS )
21SUBR CALCXY O C MVI,
31NOWR Y A LDX, MEMR Y CMPX, O<>, IF, C INR, A MEMR Y STX, THEN,
41NOWC Y A LDX, MEMC Y CMPX, O<>, IF, C INR, A MEMC Y STX, THEN,
51NOWD Y A LDX, MEMD Y CMPX, O<>, IF, C INR, A MEMD Y STX, THEN,
61DISTANCE 1+ Y A LDX, A B MOV, MEMDIST Y CMPX, O<>, IF,
71C INR, A MEMDIST Y STX, THEN,
81C A MOV, A ANA, O=, IF,
91VX Y E LDX, VX 1+ Y D LDX,
101VY Y L LDX, VY 1+ Y H LDX,
111RET,
121THEN, VBSUPDATE VLOGICSTAT Y SETX,
131--->
141
151
```

BLK= 2

```
01( MORE CUTE CALCULATIONS )
11B A MOV,
21BASEX Y L LDX, BASEX 1+ Y H LDX, DELTAX Y E LDX,
31DELTAX 1+ Y D LDX, HOTMPY CALL, L VX Y STX, H VX 1+ Y STX,
41H PUSH,
51BASEY Y L LDX, BASEY 1+ Y H LDX, DELTAY Y E LDX,
61DELTAY 1+ Y D LDX, HOTMPY CALL, L VY Y STX, H VY 1+ Y STX,
71D POP,
81RET,
91--->
101
111
121
131
141
151
```

FILE = VR

BLK= 3

```
0:( SET BASE POSITION )
1:( IN DE=ROW,COL )
2:SUBR SETBASEPOS B PUSH, D PUSH,
3:INBX C MVI, node^ CALL, M E MOV, H INX, M D MOV, H INX,
4:M C MOV, H INX, M B MOV,
5:E BASEX Y STX, D BASEX 1+ Y STX,
6:E VX Y STX, D VX 1+ Y STX,
7:C BASEY Y STX, B BASEY 1+ Y STX,
8:C VY Y STX, B VY 1+ Y STX,
9:D POP, B POP, RET,
10:SUBR FREEZEBASE A XRA,
11:A DISTANCE Y STX, A DISTANCE 1+ Y STX,
12:( A ACCDIST Y STX, A ACCDIST 1+ Y STX, )
13:A DELTADIST Y STX, A DELTADIST 1+ Y STX, RET,
14:-->
15:
```

BLK= 4

```
0:( ROUTINE TO ESTABLISH NEW BASE POSITIONS AND DELTAS )
1:( FIRST A SIGN ROUTINE )
2:SUBR SGNA A ANA, O A MVI, RP, A DCR, RET,
3:SUBR NEWPATH
4:NOWR Y D LDX, NOWC Y E LDX,
5:SETBASEPOS CALL, NOWD Y A LDX, MPLO ADI, A C MOV,
6:node^ CALL, M A MOV, A MAXDIST Y STX, S D LXI, D DAD,
7:M A MOV, A DELTAX Y STX, SGNA CALL, A DELTAX 1+ Y STX,
8:D DAD, M A MOV, A DELTAY Y STX, SGNA CALL, A DELTAY 1+ Y STX,
9:RET,
10:-->
11:
12:
13:
14:
15:
```

BLK= 5

```
0:( ROUTINE TO CAUSE OBJECT TO ARRIVE AT A NEW POSITION )
1:SUBR ARRIVE DI,
2:NOWR Y D LDX, NOWC Y E LDX, NOWD Y C LDX,
3:move#node CALL, D NOWR Y STX, E NOWC Y STX,
4:SETBASEPOS CALL, FREEZEBASE CALL,
5:RET,
6:-->
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

FILE = VR

BLK= 6

```
01( DISTANCE PHASE ACCUMULATOR )
11( DISTANCE HAS BOTH DELTA AND ACCELERATION )
21( IN A= TIMEBASE TO USE )
31SUBR DISTPA TBDEST TCHGSTAT Y BITX, RNZ,
41DISTANCE Y L LDX, DISTANCE 1+ Y H LDX,
51DELTADIST Y E LDX, DELTADIST 1+ Y D LDX,
61( ACCDIST Y C LDX, ACCDIST 1+ Y B LDX, )
71BEGIN, D DAD, ( XCHG, B DAD, XCHG, ) A DCR, O=, END,
81( IF BEYOND MAX DISTANCE, SET AT MAX DISTANCE AND FLAG )
91MAXDIST Y A LDX, A ANA, O<>, IF, H A MOV, MAXDIST Y CMPX,
101CY~, IF, TBDEST TCHGSTAT Y SETX, MAXDIST Y H LDX, O L MVI,
111THEN, THEN, E DELTADIST Y STX, D DELTADIST 1+ Y STX,
121L DISTANCE Y STX, H DISTANCE 1+ Y STX,
131RET,
141-->
151
```

BLK= 7

```
01( DISTANCE VECTORING ROUTINE AND VGER VERBS )
115 C= TB-DVECT ( TVMROPT2 BIT TO ACTIVATE DISTANCE VECTORING )
21SUBR DISTVECT PSW PUSH, B PUSH,
31B A MOV, DISTPA CALL,
41CALCXY CALL, B POP, PSW POP, RET,
51SUBR NEWVECT TB-DVECT TVMROPT2 Y BITX, vect JZ,
61H PUSH, CUSVEC Y L LDX, CUSVEC 1+ Y H LDX, XTHL, RET,
71XC? NOT IFTRUE
81HEX NEWVECT 89D9 ( 8956 ) U! DECIMAL ( ***** STUFF IN LINK )
91IFEND
101CODE DVECT-OFF Y PUSHX, vaddr LIYD, TB-DVECT TVMROPT2 Y RESX,
111Y POPX, NEXT
121CODE DVECT-ON Y PUSHX, vaddr LIYD,
131DISTVECT H LXI, L CUSVEC Y STX, H CUSVEC 1+ Y STX,
141TB-DVECT TVMROPT2 Y SETX, Y POPX, NEXT
151-->
```

BLK= 8

```
01( CODE FOR TASKS TO INTERFACE TO NEW GOODIES )
11CODE ESTPOS DI, B PUSH, Y PUSHX, vaddr LIYD,
21NOWC Y E LDX, NOWR Y D LDX,
31SETBASEPOS CALL,
41FREEZEBASE CALL,
51Y POPX, B POP, NEXT
61( TRAVEL AWAY FROM NODE )
71CODE DEPART:NODE DI, B PUSH, Y PUSHX, vaddr LIYD,
81NEWPATH CALL,
91Y POPX, B POP, NEXT
101( ARRIVE NODE )
111CODE ARRIVE:NODE DI, B PUSH, Y PUSHX, vaddr LIYD,
121ARRIVE CALL,
131Y POPX, B POP, NEXT
141-->
151
```



FILE = VR

BLK= 9

```
01( REVERSE DIRECTION ROUTINE )
1|SUBR REVERSE:DIRECTION
2|NOWR Y D LDX, NOWC Y E LDX, NOWD Y C LDX,
3|move:node CALL, C A MOV, CMA, 7 ANI,
4|D NOWR Y STX, E NOWC Y STX, A NOWD Y STX,
5|NEWPATH CALL, MAXDIST Y H LDX, O L MVI,
6|DISTANCE Y E LDX, DISTANCE 1+ Y D LDX,
7|A ANA, D DSBC, L DISTANCE Y STX, H DISTANCE 1+ Y STX,
8|RET,
9|CODE RUSH:SOURCE DI, B PUSH, Y PUSHX, vaddr LIYD,
10|DISTANCE Y A LDX, DISTANCE 1+ Y ORAX, O<>, IF,
11|REVERSE:DIRECTION CALL,
12|THEN,
13|Y POPX, B POP, NEXT
14|-->
15|
```

FILE = WR

BLK= 0

```
01( VMR      SLEZR2A )
11HEX
21SUBR SLEZR2A ( does pat offset and relabs )
31 ( in- BC= magic/exp , HL= y , DE= x , IX= pattern addr )
41 ( out- HL= new vscradr , C= new vmagic )
51 invertxy? CALL, L SLAR, H RALR, L SLAR, H RALR, ( *4 for y )
61 invert? CALL,
71 H PUSH, XCHG, O X D LDX, O E MVI, ( x offset )
81 D SRAR, E RARR, D SRAR, E RARR, ( /4 for x offset )
91 MRFLOP C BIT, 0<>, IF, D DAD, ELSE, A ORA, D DSBC, THEN,
101 XTHL, ( push X+off, HL<-Y. ) 1 X D LDX, O E MVI, ( y offset )
111 MRFLIP C BIT, 0<>, IF, D DAD, H DCX,
121 ELSE, A ORA, D DSBC, THEN,
131 D POP,
141-->
151
```

BLK= 1

```
01( VMR )
11 ( y can not set here larger than 256 )
21 H A MOV, O H MVI, A L MOV, H DAD, H DAD, H DAD,
31 H DAD, D PUSH, L E MOV, H D MOV, H DAD, H DAD, ( *64 )
41 D DAD, ( *80 ) XCHG, H POP, ( x )
51 L A MOV, ( SAVE BIT CNT ) H L MOV, O H MVI, D DAD, ( x+y )
61 RLC, RLC, 3 ANI,
71 MRFLOP C BIT, 0<>, IF, NEG, 0=, IF, H DCX, THEN, THEN,
81 3 ANI, A E MOV, invert? CALL, C A MOV, FC ANI, E ORA,
91A C MOV, ( HL= screen address ) RET,
101DECIMAL -->
111
121
131
141
151
```

BLK= 2

```
01( MY OWN EASY TO USE WRITE ROUTINE )
11BV= INTERSTAT
21CODE WRITEP A XRA, INTERSTAT STA, INTOPT IN,
31X PUSHX, D POP, EXX, X POPX, B POP, H POP, yadj CALL, XTHL,
41xadj CALL, XCHG, H POP, ( HL= Y DE= X )
51SLEZR2A CALL, X INXX, X INXX, O X E LDX, X INXX, O X D LDX,
61X INXX, write CALL, EXX,
71INTOPT IN, INTERSTAT STA,
81D PUSH, X POPX, NEXT
91DECIMAL -->
101
111
121
131
141
151
```

FILE = SC

BLK= 0

```
0:( SCORING GOODIES )
1:
2:RAMMARK SLENGTH R= P1SV RAMLEN C= P1SL VARHERE C= P1SS
3:RAMMARK SLENGTH R= P2SV RAMLEN C= P2SL VARHERE C= P2SS
4:2 A= P1SCR 2 A= P2SCR
5:9 BA= AP1SCR 9 BA= AP2SCR
6: C:S:V 0 P1SS P1SL FILL 0 P2SS P2SL FILL :
7: CLEAR:SCORES 0 P1SCR ZERO 1 P1SCR ZERO
8:0 P2SCR ZERO 1 P2SCR ZERO C:S:V :
9:-->
10:
11:
12:
13:
14:
15:
```

BLK= 1

```
0:( TASK TO DISPLAY PLAYER ONES SCORE )
1: DISPP1SCR ;TASK:
2:0 P1SCR @ 1 P1SCR @ 1 AP1SCR 7 BIN->ASC
3:8 0 AP1SCR B! 48 1 AP1SCR B!
4:0 AP1SCR OSUPR
5:-160 X! 99 Y!
6:PLOP-ON
7:7 XPAND!
8:0 AP1SCR PATTERN!
9:STRING ;
10:
11: BUMPP1SCR 0 P1SCR @ 1 P1SCR @ ROT 0 D+ 1 P1SCR ! 0 P1SCR !
12:P1SV DISPP1SCR ;
13:
14:-->
15:
```

BLK= 2

```
0:( TASK TO DISPLAY PLAYER TWOS SCORE )
1: DISPP2SCR ;TASK:
2:0 P2SCR @ 1 P2SCR @ 1 AP2SCR 7 BIN->ASC
3:8 0 AP2SCR B! 48 1 AP2SCR B!
4:0 AP2SCR OSUPR
5:96 X! 99 Y!
6:PLOP-ON
7:7 XPAND!
8:0 AP2SCR PATTERN!
9:STRING ;
10: BUMPP2SCR 0 P2SCR @ 1 P2SCR @ ROT 0 D+ 1 P2SCR ! 0 P2SCR !
11:P2SV DISPP2SCR ;
12: INCSCORE PLAYERUP @ IF BUMPP2SCR ELSE BUMPP1SCR THEN ;
13:-->
14:
15:
```

FILE = SC

BLK= 3

01( TOGGLE:LIFE, DISPLAY REMAINING LIVES, AND BITE DUST )

11: TOGGLE:LIFE INITIAL-LIVES @ -2 / + 16 \*

2:90 96 ROTY1 WRITEP ;

31

41: D:R:L

5:REMAINING-LIVES @ 1- DUP IF

6:10 DO 1 TOGGLE:LIFE LOOP

7:ELSE DROP THEN ;

81

91: BITE:DUST REMAINING-LIVES 1-!

10:REMAINING-LIVES @ DUP IF 1- TOGGLE:LIFE

11:ELSE DROP 1 GAME-OVER ! STOPme 1+B! THEN ;

121

131-->

141

151

FILE = NGM

BLK= 0

```
01( NEW CONFLICT CHECKER IN: DE=R, C B=D OUT: A= FLAG )
11DATA CONCM 1 B, 0 B, 1 B, 0 B, 0 B, 6 B, 0 B, 6 B,
215 B, 0 B, 7 B, 0 B, 0 B, 0 B, 0 B, 2 B,
31SUBR CONFLICT? B PUSH, 0 B MVI, CONCM H LXI, B DAD,
41M A MOV, A ANA, 0=, IF, B POP, RET, THEN,
51D PUSH, H PUSH, A C MOV, move:node CALL,
61H POP, 8 B LXI, B DAD, M A MOV, MPLO ADI, A C MOV,
71node^ CALL, M A MOV, D POP, B POP, RET,
81CODE CONFLICT:CHECK EXX, B POP, H POP, D POP, L D MOV,
91CONFLICT? CALL, A L MOV, 0 H MVI, H PUSH, EXX, NEXT
101
111( CHECK FOR LEGAL NODE )
121( D= ROW, E= COL RETURNS CY SET IF LEGAL COMBO )
131SUBR movecheck
141D A MOV, NROWS CPI, RNC, E A MOV, NCOLS CPI, RET, -->
151
```

BLK= 1

```
01( VARIABLES FOR MATRIX GENERATOR )
11V= GMRC V= GMD V= GMNRC
21V= RCX V= RCY V= NRCX V= NRCY
31-->
41
51
61
71
81
91
101
111
121
131
141
151
```

BLK= 2

```
01( ADD PATH ROUTINE )
11SUBR addpath GMRC SDED, C A MOV, GMD STA, ( STUFF STUFF )
21MPLO ADI, A C MOV, node^ CALL, M A MOV, A ANA, RNZ,
31GMD LDA, A C MOV, move:node CALL, GMNRC SDED, ( SET NEW R, C )
41movecheck CALL, RNC,
51GMRC LDED, CONFLICT? CALL, A ANA, RNZ,
61TOTAL-PATHS LHLD, H INX, TOTAL-PATHS SHLD, ( BUMP PATHS )
71( COMPUTE DISTANCES AND DELTAS )
81NODX C MVI, GMRC LDED, node^ CALL,
91M E MOV, H INX, M D MOV, H INX, RCX SDED,
101M E MOV, H INX, M D MOV, RCY SDED,
111GMNRC LDED, node^ CALL,
121M E MOV, H INX, M D MOV, H INX, NRCX SDED,
131M E MOV, H INX, M D MOV, NRCY SDED,
141-->
151
```

FILE = NGM

BLK= 3

```
01( COMPUTE DISTANCE )
11RCY LHL D, A ANA, D DSBC, L A MOV, SQUARE CALL, H PUSH,
21NRCX LDED, RCX LHL D, A ANA, D DSBC, L A MOV,
31SQUARE CALL, D POP, D DAD, sqrt CALL, A B MOV, ( B= DIST )
41GMRC LDED, GMD LDA, MPLO ADI, A C MOV, node^ CALL, B M MOV,
51#CON C MVI, node^ CALL, M INR,
61GMD LDA, CMA, 7 ANI, MPLO ADI, A C MOV,
71GMNRC LDED, node^ CALL,
81B M MOV, #CON C MVI, node^ CALL, M INR, RET,
91CODE ADD:PATH EXX, B POP, H POP, D POP, L D MOV,
10!addepath CALL, EXX, NEXT
11!-->
12!
13!
14!
15!
```

BLK= 4

```
01( ASSM CONNECTIVITY MARKER )
11BV= MAKCON
21F= MRPT F= MCLP F= MDLP F= NOSH F= NXTRC
31CODE MARK:CONNECTIVITY <ASSEMBLE EXX,
41LABEL MRPT A XRA, MAKCON STA, O D LXI,
51LABEL MCLP CONFLG C MVI, node^ CALL, M A MOV, A ANA,
61NXTRC JRNZ, ( SKIP IF ALREADY CONNECTED )
71MPLO CONFLG - B LXI, B DAD, O B MVI, ( B= DIR CTR )
81LABEL MDLP M A MOV, A ANA, NOSH JRZ, ( KICKOUT NOT REL )
91B C MOV, H PUSH, D PUSH,
10!move:node CALL, ( GOTO NEIGHBOR )
11!CONFLG C MVI, node^ CALL, D POP, M A MOV, H POP,
12!A ANA, ( IS NEIGHBOR MARKED? ) NOSH JRZ,
13!CONFLG C MVI, node^ CALL, 1 A MVI, A M MOV, MAKCON STA,
14!TOTAL-CONNECTS LHL D, H INX, TOTAL-CONNECTS SHLD,
15!NXTRC JMPR, -->
```

BLK= 5

```
01( TRY THE NEXT DIRECTION )
11LABEL NOSH B INR, H INX, B A MOV, 8 CPI, MDLP JRNZ,
21( GOTO NEXT GROTTO )
31LABEL NXTRC E INR, E A MOV, NCOLS CPI, MCLP JRNZ, O E MVI,
41D INR, D A MOV, NROWS CPI, MCLP JRNZ,
51( KEEP SCANNING UNTIL THANGS STABILIZED )
61MAKCON LDA, A ANA, MRPT JRNZ, EXX, NEXT
7!ASSEMBLE>
8!-->
9!
10!
11!
12!
13!
14!
15!
```

FILE = GM.

BLK= 0

```
0:( CONNECTIVITY TESTING )
1: ZAM BKGV vaddr ! NCOLS 0 DO NROWS 0 DO J I
2:COMP:XY J I NODY NODE! J I NODX NODE! LOOP LOOP ;
3: N:C CONFLG NODE^ BONE ;
4: T:C CONFLG NODEB@ : -->
5:-->
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

BLK= 1

```
0:( CONNECT INDICATED ZONES TOGETHER )
1: CRND DUP 0= IF 5 RND ELSE DUP NCOLS 1- = IF 5 RND 3 +
2:ELSE 8 RND THEN THEN ;
3: ADD:ANOTHER TOTAL-PATHS @ BEGIN NCOLS 2 - RND 1+
4:NROWS 2 - RND 1+ CRND ADD:PATH DUP TOTAL-PATHS @
5:◇ END DROP ;
6: MAKE:MAZE CLEAR:NODEMAT ZAM
7:1 TOTAL-CONNECTS !
8:NCOLS 2 - RND 1+ DUP START-COL ! 0 N:C
9:NCOLS 2 - RND 1+ STOP-COL !
10:NCOLS 0 DO NROWS 0 DO J I CRND ADD:PATH LOOP LOOP
11:BEGIN
12:1 ( INIT ) NCOLS 0 DO NROWS 0 DO J I #CON NODEB@ 2 < IF
13:J I CRND ADD:PATH DROP 0 THEN LOOP LOOP END
14:BEGIN MARK:CONNECTIVITY TOTAL-CONNECTS @ 1 = WHILE
15:START-COL @ 0 CRND ADD:PATH REPEAT -->
```

BLK= 2

```
0:( KEEP COOKING UNTIL EVERYONES CONNECTED )
1:BEGIN
2:NCOLS 0 DO NROWS 0 DO J I T:C NOT IF
3:J I CRND ADD:PATH THEN LOOP LOOP
4:MARK:CONNECTIVITY TOTAL-CONNECTS @ NNODES =
5:END
6:4 GAME# B@ 4 MIN - 4 * DUP IF 0 DO ADD:ANOTHER LOOP
7:ELSE DROP THEN ;
8:
9:( ARE WE IN THE START CHAMBER )
10: START:CHAMBER?
11:2DUP START-ROW = IF START-COL @ = IF 2DROP 0 ELSE 1 THEN
12:ELSE DROP 1 THEN ;
13:-->
14:
15:
```

FILE = LD

BLK= 0

```
01( **** LOCAL DISTANCE **** )
11( LOCAL DISTANCE ROUTINE )
21( THIS ROUTINE COMPUTES THE DISTANCE BETWEEN TWO OBJECTS )
31( IN: IX= FOLLOWER IY= LEADER OUT: A=DIST, B= REV FLAG )
41F= DIFB F= TRYM F= SAMD F= INFIN
51SUBR LDIST <ASSEMBLE
61NOWC X E LDX, NOWR X D LDX,
71( DOES CI=CO AND RI=RO ? )
81E A MOV, NOWC Y CMPX, TRYM JRNZ,
91D A MOV, NOWR Y CMPX, TRYM JRNZ,
101( ME AND HIM BOTH HAVE SAME ORIGIN )
111( ARE WE ON THE SAME BRANCH? )
121NOWD X A LDX, NOWD Y CMPX, DIFB JRNZ,
131( YES SIR - WE ARE ON SAME BRANCH )
141DISTANCE 1+ Y A LDX, DISTANCE 1+ X SUBX, 0 B MVI, BABS JMP,
151-->
```

BLK= 1

```
01( WE ARE ON DIFERENT BRANCHES OF THE SAME ORIGIN )
11LABEL DIFB DISTANCE 1+ Y A LDX,
21DISTANCE 1+ X ADDX, 1 B MVI, BABS JMP,
31LABEL TRYM NOWD X C LDX, H PUSH, move#node CALL, ( TO DEST )
41H POP, MAXDIST X A LDX, DISTANCE 1+ X SUBX, ( REVERSE DIST )
51A B MOV, ( AND SAVE IT IN B )
61D A MOV, NOWR Y CMPX, INFIN JRNZ,
71E A MOV, NOWC Y CMPX, INFIN JRNZ,
81C A MOV, CMA, 7 ANI, NOWD Y CMPX, SAMD JRZ,
91( I AM ON A PATH LEADING ME TO OTHERS ORIGIN )
101B A MOV, DISTANCE 1+ Y ADDX, 0 B MVI, BABS JMP,
111( I AM ON COMPLEMENTARY PATH THAT OBJECT IS ON )
121LABEL SAMD DISTANCE 1+ Y A LDX, B SUB, 1 B MVI, BABS JMP,
131( OBJECTS ARE FARTHER THEN WE CAN EASILY DETERMINE )
141LABEL INFIN 127 A MVI, RET,
151ASSEMBLE> -->
```

BLK= 2

```
01( DISTANCE ROUTINE FOR LIST REFORMER TO USE )
11( IF IT GETS INFINITY BACK IT WILL TRY SWAPPING X AND Y )
21
31SUBR LRDIST LDIST CALL, ( TRY IT ONE WAY )
41127 CPI, RNZ, ( RETURN IF NON INFINITE )
51( ITS INFINITE SO TRY IT THE OTHER WAY AROUND )
61X PUSHX, XTIY, X POPX, LDIST CALL,
71( BUT SWITCH BACK TO OLD POINTER SCAM BEFORE GOING HOME )
81X PUSHX, XTIY, X POPX, RET,
91-->
101
111
121
131
141
151
```



FILE = LD

BLK= 3

```
01( NEW FINDCLOSE ROUTINE )
11DECIMAL
21F= SRCL F= FCLD
31SUBR FINDCLOSE <ASSEMBLE
41O HOSTAB H LXI, EXX, 127 C MVI, EXX,
51LABEL SRCL M E MOV, H INX, M D MOV, H INX, D A MOV, E ORA,
61FCLD JRZ, D PUSH, X POPX, ASSMSV X A LDX, ASNOT CPI,
71SRCL JRNZ, HOSSV X A LDX, HSATP CPI, SRCL JRNZ,
81LDIST CALL, EXX, C CMP, CY, IF, A C MOV,
91X PUSHX, H POP, EXX, B A MOV, EXX, A B MOV, THEN,
101EXX, SRCL JMPR,
111LABEL FCLD EXX, RET,
121ASSEMBLE>
131-->
141
151
```

BLK= 4

```
01( CHECK FINDCLOSE, AND IF FOUND LIGHT UP FOLLOWER )
11SUBR LOOKFOLLOWER ( SEARCH LIST ) FINDCLOSE CALL,
21C A MOV, MAXASSM CPI, ( IS FOLLOWER CLOSE ENUF? )
31RNC, ( KICKOUT IF TOO FAR AWAY )
41DISPF Y CMPX, RC, ( OR TOO CLOSE )
51H PUSH, X POPX, ( IX= FOLLOWER )
61Y PUSHX, D POP, ( DE= LEADER )
71( LINK HER IN ) L BEHIND Y STX, H BEHIND 1+ Y STX,
81E AHEAD X STX, D AHEAD 1+ X STX, ASSIM ASSMSV X MVIX,
91DELTADIST Y A LDX, A DELTADIST X STX,
101DELTADIST 1+ Y A LDX, A DELTADIST 1+ X STX,
111B A MOV, A ANA, RZ, ( NEED WE REVERSE FOLLOWER? )
121D PUSH, H PUSH, Y POPX, REVERSE:DIRECTION CALL, Y POPX, RET,
131SUBR LOOKASS BEHIND Y A LDX, BEHIND 1+ Y ORAX, RNZ, B PUSH,
141D PUSH, H PUSH, X PUSHX, LOOKFOLLOWER CALL,
151X POPX, H POP, D POP, B POP, RET, -->
```

FILE = OT

BLK= 0

```
01( CHECK FOR ONTOP )
11F= ONTL
21SUBR ONTOP? <ASSEMBLE
31O HOSTAB H LXI, O C MVI,
41LABEL ONTL M E MOV, H INX, M D MOV, H INX,
51D A MOV, E ORA, RZ,
61D PUSH, X POPX, HOSSV X A LDX, HSATP CPI, ONTL JRNZ,
71B PUSH, LRDIST CALL, B POP, ONTOPLMT CPI, CY, IF,
811 C MVI, THEN, A B MOV, 127 CPI, O<>, IF,
91DIST-1 X SUBX,
101O=, IF, 1 C MVI, ELSE, O<, IF, 1 C MVI, THEN, THEN, THEN,
111B DIST-1 X STX, ONTL JMPR,
121ASSEMBLE>
131-->
14!
15!
```

BLK= 1

```
01( PLAYERS INTERRUPT LEVEL ONTOP CHECKER )
11SUBR PILOTR
21ASSMSV Y A LDX, A ANA, O<>, IF,
31ONTOP? CALL, C A MOV, A ANA, RNZ,
41ASCOOL ASSMSV Y MVIX, ( CLEAR ONTOP STATE )
51THEN, LOOKASS CALL, ( CHECK MY ASS )
61RET,
71SUBR PILOTC X PUSHX, PILOTR CALL, X POPX, RET,
81-->
9!
10!
11!
12!
13!
14!
15!
```

BLK= 2

```
01( PROPOGATE LEADERS DELTA DOWN THRU LIST )
11( IY= LEADERS VECTOR )
21F= CDLP SUBR COPYDELTS <ASSEMBLE
31BEHIND Y E LDX, BEHIND 1+ Y D LDX,
41LABEL CDLP
51D A MOV, E ORA, RZ, D PUSH, X POPX,
61L DELTADIST X STX, H DELTADIST 1+ X STX,
71BEHIND X E LDX, BEHIND 1+ X D LDX, CDLP JMPR,
81ASSEMBLE>
91-->
10!
11!
12!
13!
14!
15!
```

FILE = 0T

BLK= 3

```
01( MAKE ALL MY FRIENDS HALT RIGHT NOW )
11F= EHN F= RELP
21SUBR HALTNOW <ASSEMBLE
31DI, B PUSH, D PUSH, H PUSH, X PUSHX, Y PUSHX,
41O HOSTAB H LXI, PLYRV Y LXIX,
51LABEL RELP M E MOV, H INX, M D MOV, H INX,
61D A MOV, E ORA, EHN JRZ, D PUSH, X POPX,
71HOSSV X A LDX, HSATP CPI, RELP JRNZ,
81A XRA, A BEHIND X STX, A BEHIND 1+ X STX,
91A AHEAD X STX, A AHEAD 1+ X STX,
101A DELTADIST X STX, A DELTADIST 1+ X STX,
111ASNOT ASSMSV X MVIX,
121LRDIST CALL, A DIST-1 X STX, RELP JMPR,
131LABEL EHN A XRA, A BEHIND Y STX, A BEHIND 1+ Y STX,
141Y POPX, X POPX, H POP, D POP, B POP, ASONTOP A MVI,
151ASSMSV PLYRV + STA, RET, ASSEMBLE> -->
```

BLK= 4

```
01( INTERFACES TO THE TERSE WORLD )
11
21CODE PROPDeltas DI, X PUSHX, Y PUSHX, B PUSH,
31vaddr LIYD,
41DELTADIST Y L LDX, DELTADIST 1+ Y H LDX,
51COPYDELTS CALL,
61B POP, Y POPX, X POPX, NEXT
71-->
81
91
101
111
121
131
141
151
```

FILE = HF

BLK= 0

```
01( INTERFACES TO THE TERSE WORLD )
11CODE JOIN:LINE DI, X PUSHX, Y PUSHX, B PUSH,
21VADDR LIYD, HSATP HOSSV Y MVIX, PLYRV Y LXIX,
31HALTNOW CALL,
41B POP, Y POPX, X POPX, NEXT
51
61-->
71
81
91
101
111
121
131
141
151
```

BLK= 1

```
01( ASSIMULATED NODE ROUTINE )
11F= GOHM F= VIRG
21SUBR HASSIM <ASSEMBLE DI, PSW PUSH,
31DISTVECT CALL,
41LOOKASS CALL,
51VIRGIN Y A LDX, A ANA, O<, IF, 0 VIRGIN Y MVIX, VIRG JMFR,
61THEN,
71( AM I AT THE END OF THIS PATH? )
81TBDEST TCHGSTAT Y BITX, GOHM JRZ, ( NO - KICKOUT )
91-->
101
111
121
131
141
151
```

BLK= 2

```
01( MORE )
11LABEL VIRG
21X PUSHX, H PUSH, D PUSH, B PUSH, ( GRAB FARMS FROM LDR )
31NOWR B LXI, Y PUSHX, H POP, B DAD, XCHG,
41AHEAD Y L LDX, AHEAD 1+ Y H LDX, ( HL= FL )
51H PUSH, X POPX,
61B DAD, POSLEN B LXI, LDIR,
71( SET HOS DISTANCE TO N UNITS LESS THAN LEADER )
81DISTANCE 1+ X A LDX, DISPF X SUBX, O<, IF, A XRA, THEN,
91A DISTANCE 1+ Y STX, A XRA, A DISTANCE Y STX,
101TBDEST TCHGSTAT Y RESX, ( DON'T ALARM TERSE )
111B POP, D POP, H POP, X POPX,
121LABEL GOHM PSW POP, RET, ASSEMBLE> -->
131
141
151
```

FILE = HF

BLK= 3

```
01( FOLLOW MONSTER ROUTINE )
11SUBR MONF DI, B PUSH,
21Y PUSHX, H POP, NOWR B LXI, B DAD, XCHG,
31SNATCHER Y L LDX, SNATCHER 1+ Y H LDX, B DAD,
41SNATLEN B LXI, LDIR, A XRA, A DELTADIST Y STX,
51A DELTADIST 1+ Y STX,
61CALCXY CALL,
71B POP, PSW POP, RET,
81-->
91
101
111
121
131
141
151
```

BLK= 4

```
01( SPECIAL MASTER VECTORING ROUTINE FOR HOSTAGES )
11
21SUBR H1V PSW PUSH,
31H0SSV Y A LDX, HSATM CPI, MONF JRZ,
41ASSMSV Y A LDX, A ANA,
510<>, IF, PSW POP, HASSIM JMP,
61THEN, PSW POP, DISTVECT JMP,
71
81CODE HVECT-ON Y PUSHX, vaddr LIYD,
91H1V H LXI, L CUSVEC Y STX, H CUSVEC 1+ Y STX,
101TB-DVECT TVMROPT2 Y SETX, Y POPX, NEXT
111-->
121
131
141
151
```

FILE = LFN

BLK= 0

```
0!( LOOK FOR NEARBY THANGS )
1!( HL= R,C IX= SUBJ RET Z IF NEAR, NZ IF NOT )
2!SUBR NEARBY? NOWR X D LDX, NOWC X E LDX,
3!D A MOV, H CMP, 0=, IF, E A MOV, L CMP,
4!RZ, THEN,
5!DISTANCE 1+ X A LDX, A ANA, 0=, IF, A INR, RET, THEN,
6!NOWD X C LDX, H PUSH, move:node CALL, H POP,
7!D A MOV, H CMP, RNZ, E A MOV, L CMP, RET,
8!
9!( NEARBY LIST -- HL(= TARG HL= LIST RET Z= NONE NZ= FOUND )
10!SUBR NEARBYLIST M E MOV, H INX, M D MOV, H INX,
11!D A MOV, E ORA, RZ, D PUSH, X POPX, EXX,
12!NEARBY? CALL, EXX, NEARBYLIST JRNZ,
13!A MVI, A ANA, RET,
14!-->
15!
```

BLK= 1

```
0!( CODE ROUTINE TO DO NEARBY CHECK )
1!( C R LIST MTC? --- T )
2!CODE MTC? H POP, ( HL= LIST )
3!EXX, D POP, H POP, E H MOV, EXX, ( R,C )
4!X PUSHX, NEARBYLIST CALL, 0 H LXI, 0=, IF, H INX, THEN,
5!X POPX, H PUSH, NEXT
6!
7!DATA PCONFT MONV1 , MONV2 , MONV3 , MONV4 , HOSV1 , HOSV2 ,
8!HOSV3 , HOSV4 , TRSV1 , TRSV2 , TRSV3 , TRSV4 , TV1 , 0 ,
9!
10! : NOBODY:HOME:YET? 2DUP PCONFT MTC? IF 1 ELSE 2DROP 0 THEN :
11!-->
12!
13!
14!
15!
```

FILE = T

BLK= 0

```
0:( PLACE TREASURE IN MAZE )
1:TABLE T/M TRSV1 , TRSV2 , TRSV3 , TRSV4 , 0 ,
2:TABLE T/I THESTAR , THESYM , THEJEWEL , THEFLOWER , 0 ,
3:-->
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

BLK= 1

```
0:( TASK FOR A HUNK OF TREASURE )
1:
2: TRS-T :TASK: 20 RND TIMER!--ON WAIT
3:( MAKE SELF APPEAR )
4:ESTPOS
5:MYFACE V@ ANIM! 1STWRITE
6: XOR--ON ZERODXDYAXAY
7:10 TIMEBSCALE!
8:SELF MYFLAG V^ FLAG!--ON GO DI ( TREA-S ) ZEROTIMEB
9:TREA-S 2000 INCSCORE NULPAT ANIM! 1 TIMER!--ON GO ;
10:-->
11:
12:
13:
14:
15:
```

BLK= 2

```
0:( PLACE TREASURE IN MAZE )
1:V= THESPOT
2: HIDE:PEICE THESPOT ! BEGIN BEGIN
3:INCOLS RND NROWS RND START:CHAMBER? END
4:NOBODY:HOME:YET? END
5:2DUP THESPOT @ NOWR OVB! THESPOT @ NOWC OVB!
6:THESPOT @ ROLL >TREASURE NODE! THESPOT @ TRS-T ;
7: HIDE:TREASURE TOTAL-JEWELS 0 DO
8:I T/I @ I T/M @ MYFACE OV!
9:I T/M @ HIDE:PEICE LOOP ;
10: TREASURE:CHECK PUSH:CCR >TREASURE NODE@ DUP IF
11:DUP MYTYPE OVB@ T-TYP = IF
12:( JEWELS-REVEALED 1+! ) THEN 1 SWAP MYFLAG OVB!
13:0 PUSH:CCR >TREASURE NODE! ELSE DROP THEN ;
14:S
15:
```

FILE = 'RS

BLK= 0

```
0:( ROUTE SEARCH ROUTINE )
1:( VISITED MATRIX GOODIES )
2:SUBR VIS? H PUSH, B PUSH, Y PUSHX, H POP, VISMAT B LXI, B DAD,
3: E C MOV, B DAD, D A MOV, BIT^ CALL, M ANA, B POP, H POP, RET,
4:SUBR SETVIS H PUSH, B PUSH, Y PUSHX, H POP, VISMAT B LXI,
5: B DAD, E C MOV, B DAD, D A MOV, BIT^ CALL, M ORA, A M MOV,
6: B POP, H POP, RET,
7:( CLEAR OUT VIS BITMATRIX )
8:SUBR ZAPVIS B PUSH, H PUSH, VISMAT B LXI, Y PUSHX, H POP,
9: B DAD, NCOLS DO, O M MVI, H INX, LOOP, H POP, B POP, RET,
10:-->
11:
12:
13:
14:
15:
```

BLK= 1

```
0:( GENERATE TREE ENTRYS FOR ONE ENTRY )
1:IF= RUGLP
2:SUBR GENTE <ASSEMBLE MPLO C MVI, node^ CALL, H PUSH, 8 B MVI,
3: LDAR, 7 ANI, A C MOV,
4:BEGIN, H POP, H PUSH, B A MOV, O B MVI, B DAD, A B MOV,
5: M A MOV, A ANA, O<>, IF, D PUSH, move:node CALL,
6:VIS? CALL, O=, IF, ( GENERATE NODE )
7:SETVIS CALL,
8:MYBOSS Y A LDX, A TPL X STX, MYBOSS 1+ Y A LDX, A TPL 1+ X STX,
9: E TC X STX, D TR X STX, C TD X STX,
10:TREECK Y L LDX, TREECK 1+ Y H LDX, FORKETH CALL, ( END CHECK? )
11:TEL D LXI, D DADX,
12:THEN, D POP, THEN, C A MOV, A INR, 7 ANI, A C MOV, LOOP, H POP,
13:RET,
14:ASSEMBLE>
15:-->
```

BLK= 2

```
0:( ADVANCE TREE ONE DEPTH DOWN )
1:SUBR ADVT MYBOSS Y L LDX, MYBOSS 1+ Y H LDX,
2: H INX, H INX, M E MOV, H INX, M D MOV,
3:GENTE CALL, MYBOSS Y L LDX, MYBOSS 1+ Y H LDX,
4:TEL D LXI, D DAD, M E MOV, H INX, M D MOV,
5: D INX, D A MOV, E ORA, O=, IF, H INX, ELSE, H DCX, THEN,
6: L MYBOSS Y STX, H MYBOSS 1+ Y STX, ADVT JRNZ,
7:-1 X O MVIX, X INXX, -1 X O MVIX, X INXX, RET,
8:-->
9:
10:
11:
12:
13:
14:
15:
```



FILE = RS

BLK= 3

```
0:( FIND PATH ROUTINE )
1:( BC=TARGET R,C DE= NOWR,NOWC HL= ENDCHK IY= TREE RAM )
2:CODE STARTSEARCH X PUSHX, D POP, Y PUSHX, H POP, EXX,
3:H POP, vaddr LIYD, ZAPVIS CALL,
4:A XRA,
5:A FNDPTR Y STX, A FNDPTR 1+ Y STX,
6:A MYBOSS Y STX, A MYBOSS 1+ Y STX,
7:NOWR Y D LDX, NOWC Y E LDX,
8:L TREECK Y STX, H TREECK 1+ Y STX,
9:Y PUSHX, X POPX, TREES B LXI, B DADX,
10:X PUSHX, GENTE CALL, H POP,
11:L MYBOSS Y STX, H MYBOSS 1+ Y STX,
12:-1 X 0 MVIX, X INXX, -1 X 0 MVIX, X INXX,
13:X PUSHX, D POP, E FRONTIER Y STX, D FRONTIER 1+ Y STX,
14:EXX, D PUSH, X POPX, H PUSH, Y POPX, NEXT -->
15:
```

BLK= 4

```
0:( MORE PATH FINDER )
1:IF= TREELP F= SCANBK F= SCAN1
2:SUBR BANGTREE <ASSEMBLE
3:FRONTIER Y E LDX, FRONTIER 1+ Y D LDX, D PUSH, X POPX,
4:FNDPTR Y L LDX, FNDPTR 1+ Y H LDX,
5:L A MOV, H ORA, SCAN1 JRNZ, ADVT CALL,
6:X PUSHX, D POP, E FRONTIER Y STX, D FRONTIER 1+ Y STX,
7:A XRA, RET,
8:-->
9:
10:
11:
12:
13:
14:
15:
```

BLK= 5

```
0:( MORE )
1:LABEL SCAN1 O B LXI,
2:LABEL SCANBK M E MOV, C M MOV, H INX,
3:M D MOV, B M MOV, H DCX, H B MOV, L C MOV,
4:E A MOV, D ORA,
5:O<>, IF, XCHG, SCANBK JMPR, THEN, 1 A MVI, A ANA, RET,
6:ASSEMBLE>
7:
8:CODE LOOKAHEAD Y PUSHX, D POP, X PUSHX, H POP, EXX,
9:vaddr LIYD, BANGTREE CALL, O=, IF,
10:O H LXI, ELSE, H PUSH, 1 H LXI, THEN, H PUSH,
11:EXX, H PUSH, X POPX, D PUSH, Y POPX, NEXT
12:
13:-->
14:
15:
```

FILE = RS

BLK= 6

```
0:( ROUTINE TO FIND BEST PATH TOWARDS TARGET )
1:( CHECK ROUTINE - ARE WE HOME YET? )
2:SUBR BULLSEYE? INTR Y A LDX, D CMP, RNZ,
3:INTC Y A LDX, E CMP, RNZ, X PUSHX, H POP,
4:L FNDPTR Y STX, H FNDPTR 1+ Y STX, RET,
5: RECON
6:BULLSEYE? STARTSEARCH BEGIN SYNC DI
7:LOOKAHEAD END TRACKPTR V! COGO ;
8:CODE FOLLOWTRACK Y PUSHX, vaddr LIYD,
9:TRACKPTR Y L LDX, TRACKPTR 1+ Y H LDX,
10:M E MOV, H INX, M D MOV, H INX, H INX, H INX,
11:E TRACKPTR Y STX, D TRACKPTR 1+ Y STX, M L MOV, O H MVI,
12:Y POPX, H PUSH, NEXT ASSEMBLE> -->
13:
14:
15:
```

FILE = H

BLK= 0

```
0:( HOSTAGE TABLE, HOSTAGE INTERCEPT CHECKER )
1:( CHECK HOSTAGE INTERCEPT WITH MONSTERS )
2:DATA MONLIST MONV1 , MONV2 , MONV3 , MONV4 , 0 ,
3:HEX 0202 DECIMAL C= XYHOST
4:( HOSTAGES INTERCEPT CHECKER, RUNS AS HOOK )
5:SUBR HOS-MON? FREEZE? CALL, RNZ, EXX,
6:MONLIST H LXI, XYHOST B LXI, CHECK:VECTOR:LIST CALL,
7:O, IF,
8:1 MYFLAG Y MVIX, ( SET ME EATEN ) FREEZE CALL,
9:X PUSHX, D POP, E SNATCHER Y STX, D SNATCHER 1+ Y STX,
10:Y PUSHX, D POP, E MYSLAVE X STX, D MYSLAVE 1+ X STX,
11:HSATM HOSSV Y MVIX, HALTNOW CALL,
12:1 MYFLAG X MVIX, ( TELL MONSTER MOVE FLAG ) THEN,
13:EXX, RET,
14:-->
15:
```

BLK= 1

```
0:( TASK FOR A TEST HOSTAGE ) HEX 400 C= EXITVEL DECIMAL
1:( V= RECURADDR )
2: H-T ;TASK: DI H-H-D DISPF VB! H-TYP MYTYPE VB!
3:ZEROTIMEB 20 RND TIMER!--ON WAIT DI 1STWRITE
4:ESTPOS ESTVALDIR BEGIN DI 0 MYFLAG VB!
5:HOSSV VB@ HSFREE CASE DVECT-ON
6:HOS-B ANIM! XOR-ON 10 TIMEBSCALE! 0 TIMEBMAX!
7:MYFLAG V^ FLAG!--ON GO
8:ELSE HSATP CASE
9:( PRTBM TIMEBMAX! )
10:CAPT-S HOS-A ANIM! JOIN:LINE
11:1 VIRGIN VB! 0 TIMEBSCALE!
12:MYFLAG V^ FLAG!--ON HOS-MON? HOOK!--ON
13:500 INCSCORE HVECT-ON GO
14:-->
15:
```

BLK= 2

```
0:( FOLLOW MONSTER TO NEW HANGOUT )
1:ELSE HSATM CASE FREEZETH DRUG-S
2:FLAG-OFF HVECT-ON
3:HOOK-OFF
4:ZEROTIMEB
5:( FOLLOW MONSTER TO ITS TARGET POSITION )
6:BEGIN MYFLAG V^ FLAG!--ON GO DI FLAG? END
7:ESTPOS ESTVALDIR
8:UNFREEZE HSFREE HOSSV VB! ASNOT ASSMSV VB!
9:ELSE DROP THEN THEN THEN 0 END ;
10:-->
11:
12:
13:
14:
15:
```

FILE = H

BLK= 3

0:( PLACE HOSTAGES IN MAZE )

1: HIDE:HOS THESPOT ! BEGIN BEGIN

2:NCOLS RND NROWS RND START:CHAMBER? END

3:NOBODY:HOME:YET? END

4:THESPOT @ NOWR OVB! THESPOT @ NOWC OVB!

5:THESPOT @ H-T ;

6: JAIL:HOS TOTAL-HOSTAGES 0 DO

7:I HOSTAB @ HIDE:HOS LOOP ;

8:S

9:

10:

11:

12:

13:

14:

15:

FILE = R

BLK= 0

```
0:( VGS.interupt vector erase  VERASE VERASEWRITE ) <STK
1:SUBR XOR-FLIP VOXPAND Y B LDX, VOMAGIC Y C LDX,
2:VOPATH Y H LDX,
3: VOPAT Y L LDX, H INX, H INX, ( pat off set) H PUSH, X POPX,
4:  VOSCRADRH Y H LDX, VOSCRADR Y L LDX,
5:  writer JMP, ( erase it )
6:
7:
8:-->
9:
10:
11:
12:
13:
14:
15:
```

BLK= 1

```
0:( ROUTINE TO LINK TO VGER WRITE ROUTINE )
1:SUBR WRITE-LINK
2:  VBNOWRITE VLOGICSTAT Y BITX, 0=, IF, INTCPT IN, VWRITE CALL,
3:  TBINTCPT-CHK TVMROPT Y BITX, 0<>, IF, INTCPT IN,
4:  A ANA, 0<>, IF, TBINTCPT TCHGSTAT Y SETX,
5:  TBNOVECT TVMROPT Y SETX, THEN, THEN,
6:  ELSE, VBNOWRITE VLOGICSTAT Y RESX, THEN, RET, STK<> -->
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

BLK= 2

```
0:( CHECK:NEAR )
1:DATA PCON PLYRV , MONV1 , MONV2 , MONV3 ,
2:MONV4 , TV1 , TRSV1 , TRSV2 , TRSV3 , TRSV4 ,
3:HOSV1 , HOSV2 , HOSV3 , HOSV4 , 0 ,
4:-->
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

FILE = R

BLK= 3

```
0|( SPECIAL WRITE ROUTINE FOR REVEALS )
1|HEX 0C0C C= XYZONE DECIMAL
2|F= REML F= RESL F= LISTEND
3|SUBR REVEALWRITE <ASSEMBLE O H LXI, H PUSH, ( MARK STACK )
4|( Y PUSHX, H POP, CONFTAB D LXI, D DAD, )
5|PCON H LXI,
6|LABEL REML M E MOV, H INX, M D MOV, H INX, D A MOV, E ORA,
7|LISTEND JRZ, D PUSH, X POPX,
8| VBNODERASE VLOGICSTAT X BITX, REML JRNZ,
9| VOPATH X A LDX, VOPAT X ORAX, REML JRZ,
10|
11|--->
12|
13|
14|
15|
```

BLK= 4

```
0|( MORE OF SPECIAL WRITE ROUTINE FOR REVEALS )
1|XYZONE B LXI,
2|PROXIMITY-CHECK CALL, REML JRZ,
3|X PUSHX, H PUSH, Y PUSHX, X PUSHX, Y POPX, XOR-FLIP CALL,
4|Y POPX, H POP, REML JMPR,
5|LABEL LISTEND WRITE-LINK CALL,
6|LABEL RESL D POP, D A MOV, E ORA, transition JZ,
7|Y PUSHX, D PUSH, Y POPX,
8|XOR-FLIP CALL, Y POPX, RESL JMPR,
9|ASSEMBLE>
10|
11|HEX 400 C= INITIAL#LEAP
12|100 C= REVVEL 4 C= SHORTGOAL DECIMAL --->
13|
14|
15|
```

BLK= 5

```
0|( DRAW ARROWS TO REVEAL OPTIONS )
1|HEX SUBR DRAWARROWS DI, B PUSH, X PUSHX,
2|O B MVI, BEGIN,
3|B C MOV, node^ CALL, M A MOV, A ANA, O<>, IF,
4|DRAWMSK C MVI, node^ CALL, M C MOV, B A MOV,
5|BIT^ CALL, C ANA, O=, IF, B PUSH, D PUSH,
6|B C MOV, O B MVI, QUIVER H LXI, B DAD, B DAD,
7|M C MOV, H INX, M B MOV, B PUSH, X POPX,
8|NBX C MVI, node^ CALL, M E MOV, H INX, M D MOV, H INX,
9|M A MOV, H INX, M H MOV, A L MOV, 20 B LXI,
10|SLEZR2A CALL, X INXX, X INXX, O X E LDX, X INXX,
11|O X D LDX, X INXX, write CALL,
12|D POP, B POP, THEN, THEN, B INR, B A MOV, B CPI, CY~, END,
13|X POPX, B POP, RET,
14|DECIMAL --->
15|
```

FILE = R

BLK= 6

```
0:( MORE ARROWHEADED ACTIVITY )
1:BV= ARROWFLG V= ARROWRC
2:CODE ONARROWS REVEAL-ACTIVE LDA, A ANA, 0=, IF,
3:ARROWFLG LDA, A ANA, 0=, IF,
4:Y PUSHX, vaddr LIYD,
5:NOWR Y D LDX, NOWC Y E LDX, Y POPX,
6:ARROWRC SDED, DRAWARROWS CALL,
7:1 A MVI, ARROWFLG STA, THEN, THEN, NEXT
8:
9:CODE OFFARROWS ARROWFLG LDA, A ANA, 0<>, IF,
10:ARROWRC LDED, DRAWARROWS CALL,
11:A XRA, ARROWFLG STA, THEN, NEXT
12:-->
13:
14:
15:
```

BLK= 7

```
0:( HEADLIGHT REVEALER )
1:HEX : HEADLIGHT:REVEAL :TASK: DI REVEAL-ACTIVE BONE
2:NOWC PLYRV OVBE NOWC VB! NOWR PLYRV OVBE NOWR VB!
3:NOWD PLYRV OVBE NOWD VB! ESTPOS DEPART:NODE
4:MAXDIST VB@ SHORTGOAL - MAXDIST VB!
5:REVEALPAT ANIM! OC XPAND!--ON OR-ON 1STWRITE PRIBM TIMEBMAX!
6:INITIAL#LEAP DISTANCE V! REVVEL DELTADIST V! DVECT-ON
7:REVEALWRITE ZGO DI
8:-->
9:
10:
11:
12:
13:
14:
15:
```

BLK= 8

```
0:( MORE HEADLIGHT REVEALER )
1:PUSH:CCRD TEST:DRAWN NOT IF
2:REVEALED-PATHS 1+! ( INCREMENT # OF PATHS REVEALED )
3:THEN
4:PUSH:CCRD SET:DRAWN
5:ARRIVE:NODE PUSH:CCRD COM 7 AND SET:DRAWN
6:PUSH:CCR TEST:GROTTO:DRAWN NOT IF 2 REVEAL-ACTIVE B!
7:GROTTOPAT ANIM! 1STWRITE OC XPAND!--ON
8:TOTAL-REVEALED-GROTTOS 1+!
9:1 TIMER!--ON REVEALWRITE ZGO DI
10:PUSH:CCR SET:GROTTO:DRAWN THEN REVEAL-ACTIVE BZERO ;
11:DECIMAL -->
12:
13:
14:
15:
```

FILE = R

BLK= 9

```
0:( REVEAL FIRST CHAMBER )
1:HEX BV= UNROLL
2: INITIAL:REVEAL ;TASK:
3:FLYRV NOWR QVB@ NOWR VB!
4:FLYRV NOWC QVB@ NOWC VB! ESTPOS DVECT-ON
5:GROTTOPAT ANIM! 1STWRITE OC XPAND! XPAND-ON XOR-ON
6:1 TIMER!--ON REVEALWRITE ZGO
7:PUSH:CCR SET:GROTTO:DRAWN
8:18 UNROLL B!
9:BEGIN 1 TIMER!--ON WAIT UNROLL B@ DUP VERBL OUTF 4 + DUP
10:UNROLL B! ODO = END ;
11:
12:DECIMAL -->
13:
14:
15:
```



FILE = K

BLK= 0

```
0!( KEY MONITOR - WAIT FOR N CHAMBERS TO BE REVEALED )
1!<ANIM-TBL FLASHEXIT GROTTOPAT 20 NULPAT 20 TBL>
2!
3! KEY-TASK ;TASK: K-TYP MYTYPE VB! KYNONE KEY-STATUS B!
4!BEGIN 30 TIMER!--ON WAIT DI
5!TOTAL-REVEALED-GROTTOS @ KEY-THRESHOLD @ > END
6!BEGIN BEGIN
7!NCOLS RND NROWS 2- RND START:CHAMBER? END
8!NOBODY:HOME:YET? END
9!NOWR VB! NOWC VB!
10!SELF PUSH:CCR >TREASURE NODE!
11!KEY-S
12!KYSHOW KEY-STATUS B!
13!-->
14!
15!
```

BLK= 1

```
0!( KEY REVEALER )
1!ESTPOS
2!KEY1 ANIM! 1STWRITE XOR-ON
3!MYFLAG V^ FLAG!--ON DVECT-ON GO DI
4!KYOPEN KEY-STATUS B!
5!NULPAT ANIM! 1 TIMER!--ON GO
6!KEY-S
7!( NOW REVEAL EXIT CHAMBER )
8!BEGIN
9!STOP-COL B@ NOWC VB! START-ROW NOWR VB! ESTPOS
10!GROTTOPAT ANIM! PLEASE-UPDATE
11!XOR-ON XPAND-ON 8 XPAND! 30 TIMER!--ON GO DI
12!-->
13!
14!
15!
```

BLK= 2

```
0!( REVEAL THE EXIT CHAMBER )
1!GROTTOPAT ANIM! 1STWRITE 12 XPAND! XPAND-ON OR-ON
2!ESTPOS
3!1 TIMER!--ON REVEALWRITE ZGO DI
4!ESTPOS
5!FLASHEXIT ANIM!
6!1STWRITE XOR-ON XPAND-ON 8 XPAND!
7!MYFLAG V^ FLAG!--ON GO KYGONE KEY-STATUS B! ;
8!-->
9!
10!
11!
12!
13!
14!
15!
```

FILE = K

BLK= 3

```
0:( MORE EXIT REVEALER AND KEY HIDER )
1:
2: HIDE:KEY BEGIN BEGIN
3:NCOLS RND NROWS 2- RND START:CHAMBER? END
4:NOBODY:HOME:YET? END
5:2DUP TV1 NOWR OVB! TV1 NOWC OVB!
6:TV1 ROLL >TREASURE NODE! TV1 KEY-TASK ;
7:-->
8:
9:
10:
11:
12:
13:
14:
15:
```

BLK= 4

```
0:( ROUTINE TO END GAME )
1: END-GAME ;TASK:
2:0 BEHIND PLYRV OVB BEGIN DUP WHILE SWAP 5000 + SWAP
3:BEHIND OVB REPEAT DROP INCSCORE 60 TIMER!-ON WAIT
4:STOPme 1+B! NOBREAK BZERO ;
5:-->
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

FILE = P

BLK= 0

```
0:( JOYSTICK ROUTINES )
1:HEX ( BV= JOYCODE BV= JOYLAST ) ( D800 DP ! ***** )
2:DATA JOYTBL -1 B, -1 B, -1 B, -1 B, -1 B, 0 B, 5 B, -1 B,
3:-1 B, 2 B, 7 B, -1 B, -1 B, -1 B, -1 B, -1 B,
4:-1 B, 1 B, 6 B, -1 B, 3 B, 0 B, 5 B, -1 B,
5:4 B, 2 B, 7 B, -1 B, -1 B, -1 B, -1 B, -1 B,
6:( SUBR MYINTR PSW PUSH, H PUSH, 12 IN, CMA, 1F ANI,
7:JOYLAST H LXI, M CMP, A M MOV, 0<>, IF, 1F A MVI, THEN,
8:JOYCODE STA, H POP, PSW POP, SUI1 JMP, )
9:SUBR set:Joycode 12 IN, CMA, 1F ANI, A E MOV, 0 D MVI,
10:JOYTBL H LXI, D DAD, M A MOV, A ANA, RET,
11:CODE GET:JOYCODE
12:12 IN, CMA, 1F ANI, A E MOV, 0 D MVI, JOYTBL H LXI,
13:D DAD, M A MOV, A ANA, 0<>, IF, 0 H LXI, ELSE,
14:A E MOV, D PUSH, 1 H LXI, THEN, H PUSH, NEXT
15:DECIMAL -->
```

BLK= 1

```
0:( NEW SCAN ADJUSTER )
1:DATA CCWTBL 3 B, 0 B, 1 B, 5 B, 2 B, 6 B, 7 B, 4 B,
2:DATA CWTBL 1 B, 2 B, 4 B, 0 B, 7 B, 3 B, 5 B, 6 B,
3:F= scanr F= nosc
4:SUBR adj-scan <ASSEMBLE
5:H PUSH, 0 B MVI, B DAD, M A MOV, A ANA,
6:scanr JRZ, H POP, C A MOV, RET,
7:LABEL scanr CCWTBL H LXI, B DAD, M E MOV, 0 D MVI,
8:H POP, H PUSH, D DAD, M D MOV,
9:CWTBL H LXI, B DAD, C A MOV, M C MOV, H POP, B DAD,
10:A B MOV, M A MOV,
11:A ANA, 0<>, IF, D A MOV, A ANA, nosc JRNZ,
12:C A MOV, RET, THEN, D ORA, nosc JRZ, E A MOV, RET,
13:LABEL nosc B A MOV, RET,
14:ASSEMBLE>
15:-->
```

BLK= 2

```
0:( INTERRUPT LEVEL JOY MONITOR )
1:CODE ADJ-SCAN EXX, B POP, H POP,
2:adj-scan CALL, A L MOV, 0 H MVI, H PUSH, EXX, NEXT
3:BV= OBJECT-MOVING
4:F= RVRS
5:SUBR JOYCHECK <ASSEMBLE OBJECT-MOVING LDA, A ANA, RZ,
6:TBDEST TCHSTAT Y BITX, RNZ, DISTANCE 1+ Y A LDX, A ANA, RZ,
7:set:Joycode CALL,
8:0<>, IF, PLYRV ASSMSV + LDA, ASCOOL CPI, 0<>, IF,
9:PLYRV MAXDIST + LDA, DISTANCE 1+ Y SUBX, COASTZONE CPI,
10:CY~, IF,
11:0 H LXI, PLYRV DELTADIST + SHLD, THEN, THEN, RET,
12:THEN, PLAYERVELO LHL, PLYRV DELTADIST + SHLD,
13:-->
14:
15:
```

FILE = P

BLK= 3

```
01( CHECK FOR REVERSAL )
11CMA, 7 ANI, NOWD Y E LDX, E CMP, RVRS JRZ,
21O D MVI, CWTBL H LXI, D DAD, M CMP, RVRS JRZ,
31CCWTBL H LXI, D DAD, M CMP, RNZ,
41LABEL RVRS
51REVERSE: DIRECTION CALL, HALTNOW CALL,
61NOWD Y A LDX, RRC, RRC, RRC, A VANGLE Y STX, RET,
71ASSEMBLE>
81SUBR PL-M JOYCHECK CALL, PILOT CALL, RET,
91-->
101
111
121
131
141
151
```

BLK= 4

```
01( CHECK FOR PLAYER ESCAPING INTO EXIT CHAMBER )
11CODE ESCAPE? KEY-STATUS LDA, KYOPEN CPI, 0=, IF,
21B PUSH, Y PUSHX, vaddr LIYD,
31STOP-COL LDA, NOWC Y CMPX, 0=, IF,
41NOWR Y A LDX, START-ROW CPI, 0=, IF,
51( WE WIN! - SHAZAM! )
61STOPme H LXI, M INR, ( SHUTUP )
71A XRA, NOBREAK STA,
81THEN, THEN,
91Y POPX, B POP, THEN, NEXT
101-->
111
121
131
141
151
```

BLK= 5

```
01( PLAYER HOSTAGE INTERFACE JUNK )
11F= DISH
21SUBR dishos <ASSEMBLE O HOSTAB H LXI,
31LABEL DISH M E MOV, H INX, M D MOV, H INX, D A MOV, E ORA, RZ,
41XCHG, HOSSV B LXI, B DAD, M A MOV, HSATP CPI, 0=, IF,
51HSFREE M MVI, MYFLAG HOSSV - B LXI, B DAD, 1 M MVI, THEN,
61XCHG, DISH JMPR, ASSEMBLE>
71CODE DISHOS B PUSH, dishos CALL, B POP, NEXT
81CODE HALTER HALTNOW CALL, NEXT
91-->
101
111
121
131
141
151
```

FILE = P

BLK= 6

```
0!( CHECK VECTOR FOR INTERCEPT WITH OTHER VECTORS )
1!( ROUTINE TO FIND INTERCEPTORS, IF ANY )
2!( ENTRY: BC= NEARNESS X AND Y, HL= CHECKLIST ADDR )
3!( IY= SUBJECT VECTOR )
4!( RETURNS Z= NOFIND, NZ= FIND, IX= FOUND THANG )
5!F= C:UH
6!SUBR C:U:H <ASSEMBLE
7!LABEL C:UH
8!M E MOV, H INX, M D MOV, H INX, D A MOV, E ORA,
9!RZ, D PUSH, X POPX,
10!HOSSV X A LDX, HSFREE CPI, 0=, IF,
11!PROXIMITY-CHECK CALL, RNZ, THEN, C:UH JMPR,
12!ASSEMBLE>
13!-->
14!
15!
```

BLK= 7

```
0!( CHECK PLAYER INTERCEPT WITH OTHER VECTORS )
1!O C= EATEN 1 C= EATHOST
2!DATA CHECKLIST MONV1 , MONV2 , MONV3 , MONV4 , 0 ,
3!HEX 0202 DECIMAL C= XYBOUNDS
4!( PLAYERS INTERCEPT CHECKER, RUNS AS HOOK )
5!SUBR P:I:C FREEZE? CALL, RNZ, EXX,
6!CHECKLIST H LXI, XYBOUNDS B LXI, CHECK:VECTOR:LIST CALL,
7!O<>, IF, 1 A MVI, PLAYERDEAD STA, FREEZE CALL,
8!EATEN FLAGCODE X MVIX, A MYFLAG X STX, ( SET EATEN FLAG )
9!3 A MVI, 4 OUT, ELSE,
10!( ANY HOSTAGE ABOUT? )
11!O HOSTAB H LXI, XYBOUNDS B LXI, C:U:H CALL,
12!O<>, IF, 1 MYFLAG X MVIX, HSATP HOSSV X MVIX, THEN,
13!A XRA, THEN, 4 OUT,
14!EXX, RET,
15!-->
```

BLK= 8

```
0!( CHECK VMAX SWITCH )
1!HEX
2!CODE VMAX? 0 H LXI, 12 IN, 5 A BIT, 0=, IF, H INX, THEN,
3!H PUSH, NEXT
4!
5!CODE SETVEL EXX, H POP, Y PUSHX, vaddr LIYD,
6!L DELTADIST Y STX, H DELTADIST 1+ Y STX, PLAYERVELO SHLD,
7!Y POPX, EXX, NEXT
8!DECIMAL <-->
9!
10!
11!
12!
13!
14!
15!
```

FILE = P

BLK= 9

```
01( EXPLORE-MAZE )
11: ROTUND ;TASK: DI
21H-P-D DISPF VB! ESTPOS
31ROTROTY ANIM! XOR-ON 1STWRITE PRFBM TIMEBMAX!
41BEGIN DI ONARROWS
51PUSH:CCR TEST:GROTTO:DRAWN IF GET:JOYCODE ELSE 0 THEN
61PUSH:CCR MPLO NODE^ SWAP ADJ-SCAN
71DUP NOWD VB@ COM 7 AND - IF HALTER THEN DUP NOWD VB!
81DUP 32 * VANGLE-VB!
91-->
10!
11!
12!
13!
14!
15!
```

BLK= 10

```
01( MORE PLAYER STUFF )
11PUSH:CCR ROT TEST:REL
21IE ZEROTIMER
31PUSH:CCR ROT TEST:REL
41TEST:DRAWN IF
51VMAX? IF 512 ELSE 384
61THEN ELSE 256 THEN SETVEL
71DEPART:NOBE
81OBJECT-MOVING BONE
91DEPART:NOBE
101PUSH:CCRD TEST:DRAWN NOT IF DIG-S
11:100 INCSCORE
12:REVEAL-ACTIVE B@ 2 = IF BEGIN SYNC REVEAL-ACTIVE B@ 0= END THEN
13:REVV HEADLIGHT:REVEAL SYNC DI ROTROTY ANIM! ELSE WALK-S THEN
14!-->
15!
```

BLK= 11

```
01( EXPLORE-MAZE )
11ELSE 0 SETVEL 3 TIMER!-ON
21THEN ELSE 0 SETVEL 3 TIMER!-ON THEN
31P:ITC HOOK!-ON
41PROPDELTA$ PLAYERDEAD FLAG!-ON DVECT-ON mastersur IGO DI
51OBJECT-MOVING BZERO
61FLAG? IF MELT-S ZEROTIMEB DEATHACT ANIM! BITE:DUST
710 SETVEL HALTER DISHOS
8120 TIMER!-ON GO DI ROTROTY ANIM!
91START-COL B@ NROWS 1- SET:NEW:MCCR ESTPOS
101PLAYERDEAD ZERO THEN DI
11!-->
12!
13!
14!
15!
```

FILE = P

BLK= 12

0:( YET MORE PLAYER CONTROLLER )

1:DEST? IF ARRIVE:NODE PROPDELTA

2:ESCAPE? TREASURE:CHECK DI ROTROTY ANIM! THEN

3:O END ; DECIMAL -->

4:

5:

6:

7:

8:

9:

10:

11:

12:

13:

14:

15:

FILE = IP

BLK= 0

```
01( PROCESS A HOT ROD MISSLE )
11BV= HOTFLIP
21SUBR HOTROD
31 TBMISSLE TSTAT Y BITX, ( are we ready to process )
41 RZ, ( NOT A MISSLE )
51 ( A= timebase ) mastervmr CALL,
61 VBMISWRT VLOGICSTAT Y BITX, ( time to write ? )
71 VBMISWRT VLOGICSTAT Y RESX,
81 O<>, IF, TSUR Y L LDX, TSUR 1+ Y H LDX, FORKETH CALL,
91THEN, RET,
101-->
111
121
131
141
151
```

BLK= 1

```
01<STKH
11SUBR MIS-INT ( missile interrupt test )
21 PSW PUSH, B PUSH, D PUSH, H PUSH, EXX, EXAF,
31 PSW PUSH, B PUSH, D PUSH, H PUSH, Y PUSHX, X PUSHX,
41( 12 IN, CMA, 1F ANI,
51JOYLAST H LXI, M CMP, A M MOV, O<>, IF, 1F A MVI, THEN,
61JOYCODE STA, ) ( HOT ROD THE PLAYERS VECTOR )
71HOTFLIP H LXI, M A MOV, A INR, 1 ANI, A M MOV,
81PLYRV Y LXIX, O=, IF, PL-M CALL,
91ELSE, 2 A MVI, HOTROD CALL, ( REVV Y LXIX, THEN, 2 A MVI, )
101( HOTROD CALL, )
111THEN,
121 SUI2-NP JMP,
131: MYPUP MYPUP MIS-INT SUI1V ! -1 HORCB OUTP ; STK> -->
141
151
```



FILE = M

BLK= 0

```
01( INDEXER AND VISABLE MONSTER WRITER )
11: I:M MONVBYTES * MONV1 SWAP - ;
2:
3:SUBR VISIONWRITE ( VISABLE MONSTER WRITER )
4: VBNOERASE VLOGICSTAT Y BITX, 0=, IF,
5: VOPATH Y A LDX, VOPAT Y DRAX, 0<>, IF,
6: VERASE CALL, THEN, ( don't erase if no pattern )
7: ELSE, VBNOERASE VLOGICSTAT Y RESX, THEN,
8: VBNOWRITE VLOGICSTAT Y BITX, 0=, IF, INTOPT IN, VWRITE CALL,
9: TBINTOPT-CHK TVMROPT Y BITX, 0<>, IF, INTOPT IN,
10: A ANA, 0=, IF, TBINTOPT TCHGSTAT Y SETX,
11:     TBNOVECT TVMROPT Y SETX, THEN, THEN,
12: ELSE, VBNOWRITE VLOGICSTAT Y RESX, THEN,
13: transition JMP, -->
14:
15:
```

BLK= 1

```
01( MONSTER STUFF )
1:DECIMAL
2: BANISH:MONSTER BEGIN BEGIN NCOLS RND DUP INTO VB!
3:NOWC PLYRV OVB@ - ABS 2 > END BEGIN NROWS RND DUP INTR VB!
4:NOWR PLYRV OVB@ - ABS 1 > END INTO VB@ INTR VB@ NOBODY:HOME:YET?
5: END 2DROP ;
6: MONGO INTERCEPT-ON DVECT-ON
7:VISFLAG VB@ IF MYFACE V@ ANIM! VISIONWRITE ZOO DI
8:INTERCEPT? IF 0 VISFLAG VB! THEN
9:ELSE EYEBALLS-PAT ANIM! GO DI INTERCEPT? IF 1 VISFLAG VB! THEN
10:THEN COGO ;
11: FREESLAVE DI MYSLAVE V@ IF MYSLAVE V@ MYFLAG + BONE
12:( 0 MYSLAVE V@ SNATCHER + ! )
13:0 MYSLAVE V! THEN ;
14:-->
15:
```

BLK= 2

```
01( MORE MONSTER STUFF )
1:( COMPARE POSITION IN D AND E WITH POSITION IN VECTOR )
2:SUBR compos D A MOV, NOWR Y CMPX, RNZ,
3: E A MOV, NOWC Y CMPX, RET,
4:CODE CHASEPLAYER EXX, X PUSHX, Y PUSHX,
5:PLYRV X LXIX, vaddr LIYD,
6:NOWR X D LDX, NOWC X E LDX, NOWD X C LDX,
7:move:node CALL, movecheck CALL, CY, IF,
8:compos CALL, 0=, IF, ( IF AT PLAYERS DEST, GRAB HIS SOURCE )
9:NOWR X D LDX, NOWC X E LDX, THEN,
10:D INTR Y STX, E INTO Y STX,
11:THEN, EXX, Y POPX, X POPX, NEXT
12:( GO ANYWHERE I AM NOT NOW )
13: VAMOOSE BEGIN NCOLS RND INTO VB! NROWS RND INTR VB!
14:ON:TARGET? NOT END ;
15:-->
```

FILE = M

BLK= 3

```
0:( MONSTER TASK )
1:HEX TABLE MONVEL 60 , 80 , A0 , C0 , 100 , DECIMAL
2!: RODAN? MYFACE V@ THEWAROD1 = ;
3:DECIMAL
4!: MONSTER-TASK ;TASK: DI
5:RETURN:INITIAL:POSITION
6:ESTPOS
7:MYFACE V@ ANIM! XOR-ON 1STWRITE BEGIN DI
8:ON:TARGET? IF RODAN? IF
9:1 ELSE SMARTS B@ RND THEN IF CHASEPLAYER
10:ON:TARGET? IF VAMOOSE THEN ELSE VAMOOSE THEN
11: / RECON SETCO COGO DI ZEROTIMEB
12:-->
13:
14:
15:
```

BLK= 4

```
0:THEN FOLLOWTRACK NOWD VB!
1:GAME# @ RODAN? +
2:4 MIN MONVEL @ DELTADIST V! DEPART:NODE
3:( HAVE MONSTER CRAWL ABOUT )
4:BEGIN MYFLAG V^ FLAG!-ON
5: / MONGO SETCO COGO DI
6:-->
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

BLK= 5

```
0:( BANISHMENT STUFF )
1:FLAG? IF 0 DELTADIST V!
2:BANISH:MONSTER INTC VB@ BANC B!
3:INTR VB@ BANR B!
4: / RECON SETCO COGO DI
5:0 MYFLAG VB! FLAG-OFF
6:( WANDER BACK TO WHERE MONSTER LAST CAME FROM )
7:BEGIN ESTPOS ZEROTIMEB
8:ON:TARGET? NOT IF FOLLOWTRACK NOWD VB!
9:DEPART:NODE EXITVEL DELTADIST V!
10:BEGIN / MONGO SETCO COGO DEST? END ARRIVE:NODE 0
11:ELSE 1 THEN END
12:FREESLAVE
13:UNFREEZE 1 ELSE 0 DEST? IF ARRIVE:NODE DROP 1 THEN THEN
14:END 0 END ;
15:DECIMAL -->
```

FILE = M

BLK= 6

```
0:( MONSTER MASH )
1:BTABLE MRTBL 0 B, 0 B, 2 B, 2 B,
2:BTABLE MCTBL 0 B, NCOLS 1- B, 0 B, NCOLS 1- B,
3: MONSTERMASH MONSTERCOUNT @ 0 DO I MCTBL B@ I MRTBL B@
4:I I:M SET:INITIAL:MCCR I 0= IF THEWAROD1 ELSE THESPDR
5: THEN I I:M MYFACE OV! I I:M MONSTER-TASK
6:LOOP :
7:-->
8:
9:
10:
11:
12:
13:
14:
15:
```

FILE = E

BLK= 0

```
01( PRE VGER ACTIVITY ) HEX
11XC? IFTRUE : CLMUS 0 BGMV TLENGTH FILL ;
21CODE CRAMIT OD800 H LXI, BEGIN, 0 M MVI, H INX, H A MOV,
31OFO CPI, 0=, END, NEXT
41OTHERWISE : CRAMIT ; : CLMUS ; IFEND
51: VG MYPUP DI CRAMIT SPARKLES-OFF CLEAR:SCORES ZAP:VECT
618 0 DO 8 I OUTP LOOP
714 DUP REMAINING-LIVES ! INITIAL-LIVES !
81GAME-OVER ZERO
91GAME# ZERO
101BEGIN TOTAL-PATHS ZERO REVEAL-ACTIVE BZERO ARROWFLG BZERO
111CHEAPRND 0 RND# !
121MAKE:MAZE MD
131SCRERASE
141( BLUEFILL ) -1 4000 800 FILL
151-->
```

BLK= 1

```
01( MORE EXPLORE )
11DI CLMUS MYPUP AMUSE
2118 VERBL OUTP -1 HORCB OUTP
31NOBREAK BONE ZAP:VECT
41C:S:V
51HIDE:TREASURE JAIL:HOS
61NPLAYERS ZERO PLAYERUP ZERO
71REVEALED-PATHS ZERO 1 TOTAL-REVEALED-GROTTOS !
81-->
91
101
111
121
131
141
151
```

BLK= 2

```
01( PRE VGER ACTIVITY )
11START-COL @ DUP PLYRV NOWC OVB!
21REVV NOWC OVB!
31START-ROW DUP PLYRV NOWR OVB!
41REVV NOWR OVB! PLAYERDEAD ZERO
513 GAME# @ + 4 MIN MONSTERCOUNT ! STARTEXCITE BACK-S
61GAME# @ 1+ 4 * 26 MIN KEY-THRESHOLD !
71GAME# @ 2/ 1+ SMARTS B! FREEZEFLAG BZERO
81P1SV DISPP1SCR P2SV DISPP2SCR
91BKOV INITIAL:REVEAL
101PLYRV ROTUND ( JOYV JOYSTICK-MONITOR )
111MONSTERMASH TV1 KEY-TASK
121D:R:L 8 7 OUTP
131-->
141
151
```

FILE = E

BLK= 3

```
0:( YET MORE )
1:TT GAME# 1+! NOBREAK B@ DUP 0= IF DI MYPUP 0 TVVS TVVL FILL
2:TV1 END-GAME TT THEN
3:( 8 0 DO 8 I OUTP LOOP )
4:GAME-OVER B@ OR EMUSIC END ;
5:
6:HEX
7: GAMELP BEGIN CRAMIT VG BEGIN 10 INP OFF <> END 0 END ;
8:DECIMAL -->
9:
10:
11:
12:
13:
14:
15:
```