

Complexity of Cooperation Web Site

TourExec1.1.f

```
Program AxTest
c For testings of PD tour program.
c Begun 7/19/93. Ver 1.0 begun 7/27 for nice rules as well as not nice rules
c Compile: set directory. then: RUN tourexec2 -debug -saveall -ov -r

c Changes to make:
c Add Almost-Pavlov and Almost-TFT to col rules

real Version /1.1/
c Next few lines are control parameters
integer ColType/4/ ! 1=TFT, 2=TF2T, 3=Random, 4= Pavlov
integer MoveReport/0/ ! 0= no report of moves, 1 = report moves
integer GameReport/0/ ! 0 = no report of games 1= report games
real Noise/0./ ! prob a choice will be changed
integer minRow/1/ ! normally /1/ to run all rules
integer maxRow/63/ ! normally /63/ to run all rules
integer outcome(308) ! 1=R, 2=T, 3=S, 4=P for Column
integer length(5) /63,77,151,156,308/ ! Game Lengths in Tour
integer game ! Game no. with this pair, 1 to 5
integer*4 RandomSeed !
integer Row, Rank ! Row = Rank = 1..63 for 2nd round rules
integer RowGameScore, ColGameScore ! Score in Current Game
integer Tally(4) ! tally of col's outcomes for game
integer ColOutcomeType ! 1=R, 2=T, 3=S, 4=P for Column
integer RowGameSc, ColGameSc ! Scores in one game
integer RowPairSc, ColPairSc ! Scores over 5 games
integer MoveRecord(308) ! Moves of current game
character*9 day
character*8 timenow
integer ActualTFTTourSc(63)/453,453,453,453,453, 453,453,452,453,453,
1 453,453,453,453,453, 449,453,452,450,453,
2 453,453,453,453,452, 453,446,453,449,453,
3 453,453,453,453,453, 453,453,453,452,453,
4 453,453,453,453,453, 452,453,443,422,452,
5 442,453,452,442,342, 398,377,388,438,155,
6 376,341,198/
integer IRowPairSc(63), IColPairSc(63) ! Integer total over 5 games
real AveRowPairSc(63), AveColPairSc(63) ! real, truncated
integer rowchoice, colchoice
call Date(day)
call TIME(timenow)
write(6,100) Version, day, timenow
100 format(' Ax TourExec Program Output, Version ',f6.2, '.', 1H, A10, A10)
RandomSeed = Jsecnds(0) ! uses elapsed time since midnight as random seed
c RandomSeed=66222 ! Uses fixed random number
Write(6,103) RandomSeed
103 format(' RandomSeed = ', i16)

write(6,85) noise
85 format(' Noise (per choice) = ', f8.4)
```

```

write(6, 104) ColType
104 format(' Col Type, 1=TFT, 2=TF2F, 3=Random, 4=Pavlov. Col Type = ', i3)
if (movereport=1) write(6, 105)
105 format(' Move report: 1 means R, 2 means T, 3 means S, 4 means P for column.')
if (GameReport=1) write(6,101)
101 format(' Rank Game RScore CScore #ColR #ColT #ColS #ColP')
ITotalColPoints = 0 ! Initialize Col's total points
Do 30 row= minRow,maxRow ! normally 1 to 63
rank = row
RowPairSc = 0
ColPairSc = 0

Do 20 Game = 1,5
RowGameSc = 0
ColGameSc = 0
JA = 0 ! Row's previous move, reported to column
JB = 0 ! Col's previous move, reported to row
Do 10 ColOutcomeType = 1,4
Tally(ColOutcomeType) = 0 ! Zero Col's RTSP game count
10 Continue ! End Do tallyType
Do 15 Move = 1, Length(Game)
RandomNumber = RAN(RandomSeed)
RowChoice = KRowFunction(JB,Move, RowGameSc,ColGameSc,RandomNumber,Row,JA)
if ( RAN(RandomSeed) < noise ) RowChoice = 1-RowChoice ! noise happened to Row
RandomNumber = RAN(RandomSeed)
ColChoice = KColFunction(JA,Move,ColGameSc,RowGameSc,RandomNumber,ColType,JB)
if ( RAN(RandomSeed) < noise ) ColChoice = 1 - ColChoice ! noise happened to Col
C temp test:
c Write(6, 999) Move, RowChoice, ColChoice
c999 Format(' move, rowchoice, colchoice ', 3i6)
ColOutcomeType = 1 + 2*RowChoice + ColChoice ! *check col: 1=R,2=T
Tally(ColOutcomeType) = Tally(ColOutcomeType) + 1
JA = RowChoice ! Reported to col next time
JB = ColChoice ! Reported to row next time

Select Case (ColOutcomeType)
Case (1) ! Both Get R
RowGameSc=RowGameSc+3
ColGameSc=ColGameSc+3
Case (2) ! Col Gets T
ColGameSc=ColGameSc+5
Case (3) ! Col Gets S
RowGameSc=RowGameSc+5
Case (4) ! Both Get P
RowGameSc=RowGameSc+1
ColGameSc=ColGameSc+1
End Select

MoveRecord(move)=ColOutcomeType
15 Continue ! End Do Move

C write game output
RowPairSc=RowPairSc+RowGameSc ! sum over 5 games
ColPairSc=ColPairSc+ColGameSc
if (GameReport=1) Write(6, 110) Rank, Game, RowGameSc,
1 ColGameSc, Tally(1), Tally(2), Tally(3), Tally(4)
110 format(9i6, 10i3)

```

```

    if (movereport .eq. 1) write(6, 112) (MoveRecord(ir), ir=1,length(game))
112 format(' ', 10i2, 2H, 10i2, 2H, 10i2, 2H, 10i2)
20 Continue          ! End Do Game
    if (GameReport=1) write(6, 115) RowPairSc, ColPairSc
    IRowPairSc(Row) = RowPairSc          ! total over 5 games
    IColPairSc(Row) = ColPairSc
    IColTourSc = IColTourSc + ColPairSc ! running total of col's points
115 format(' Totals over 5 games: RowPairSc= ',I7, ' ColPairSc = ', I7)
    if (GameReport=1) write (6, 120)
120 format()

30 Continue          ! End Do Row

C final report: calc tour score, write tour output

    Write(6, 135)
135 format(' Rank   RowSc   ColSc   AveRowSc AveColSc 2ndRndTFT  2ndRndTFT-Col')
    Do 40 Row = minRow,maxRow
    IRowTourPairSc = IRowPairSc(Row)/5
    IColTourPairSc = IColPairSc(Row)/5
    ITotalColPoints = ITotalColPoints + IColPairSc(Row) ! accumulate col points
    Write(6, 140) Row, IRowPairSc(Row), IColPairSc(Row),IRowTourPairSc,
        2 IColTourPairSc, ActualTFTTourSc(Row), ActualTFTTourSc(Row)-IColTourPairSc
140 format(i6, 4i8, ' ',i8,' ',i8)
40 continue          ! end final report
    TotalColPoints = ITotalColPoints ! to make floating point (total over 63*5 games)
    ColTourSc =(TotalColPoints/5 )/63 ! Ave per game over 63 pairs
    write(6, 150) ColType, ITotalColPoints, ColTourSc
150 format(' Col Type= ', i4, '. Col Pts = ', i7, ' Col"s Tour Sc = ', f7.3)
    end ! Main Program

C-----
Function KColFunction(J,M,K,L,R,IColType,JB) ! Look up col rule, return col choice
if (icoltype. eq. 1) KColFunction= KTitForTatC(J,M,K,L,R)
if (icoltype .eq. 2) KColFunction= KTF2TC(J,M,K,L,R)
if (icoltype .eq. 3) KColFunction= KRandomC(J,M,K,L,R)
if (icoltype .eq. 4) KColFunction= KPavlovC(J,M,K,L,R, JB) ! JB is own, col's prev move
return
end

C-----
Function KTitForTatC(J,M,K,L,R) ! TFT, Row Rule
KTitForTatC = J
Return
End ! TFT Col Rule

C-----
Function KTF2TC(J,M,K,L,R) ! Tit for Two Tats, Col rule
if(m .eq. 1) jold = 0
ktf2tc = 0
if ((jold .EQ. 1) .and. (j .eq. 1)) ktf2tc = 1
jold = j
Return
End ! TF2T Col Rule

C-----
Function KRandomC(J,M,K,L,R) ! Random, Row Rule
KRandomC = 0
if (R .LE. .5) KRandomC = 1
Return
End ! Random Col Rule

```

```

C -----
  Function KPavlovC(J,M,K,L,R,JB)      ! Pavlov, JB is own (Col) previous move
c   coded by Ax 7/22-3/93. Assumes C on first move.
  KPavlovC = 1
  if (J .eq. JB) KPavlovC = 0 ! coop iff other's previous choice= own previous ch
C test3
c   write(6,81) J, JB
c81 format(2i3, 'j,jb from test3')
  Return
  end
c-----

```

```

c-----
  Function KRowFunction(J,M,K,L,R,iRow,JA)      ! Look up row rule, return rowchoice
c add JA to row fcns to report their own previous move, 7/23/93
  if (irow>32 ) goto 133
  if (irow>16 ) goto 117
  if (irow>8 ) goto 109
  if (irow>4 ) goto 105
  if(irow=1) KRowFunction = K92R(J,M,K,L,R,JA)
  if(irow=2) KRowFunction = K61R(J,M,K,L,R,JA)
  if(irow=3) KRowFunction = K42R(J,M,K,L,R,JA)
  if(irow=4) KRowFunction = K49R(J,M,K,L,R,JA)
  return
105 if(irow=5) KRowFunction = K44R(J,M,K,L,R,JA)
  if(irow=6) KRowFunction = K60R(J,M,K,L,R,JA)
  if(irow=7) KRowFunction = K41R(J,M,K,L,R,JA)
  if(irow=8) KRowFunction = K75R(J,M,K,L,R,JA)
  return
109 if(irow>12) goto 113
  if(irow=9) KRowFunction = K84R(J,M,K,L,R,JA)
  if(irow=10) KRowFunction = K32R(J,M,K,L,R,JA)
  if(irow=11) KRowFunction = K35R(J,M,K,L,R,JA)
  if(irow=12) KRowFunction = K68R(J,M,K,L,R,JA)
  return
113 if(irow=13) KRowFunction = K72R(J,M,K,L,R,JA)
  if(irow=14) KRowFunction = K46R(J,M,K,L,R,JA)
  if(irow=15) KRowFunction = K83R(J,M,K,L,R,JA)
  if(irow=16) KRowFunction = K47R(J,M,K,L,R,JA)
  return
117   if (irow>24 ) goto 125
  if (irow>20 ) goto 121
  if(irow=17) KRowFunction = K64R(J,M,K,L,R,JA)
  if(irow=18) KRowFunction = K51R(J,M,K,L,R,JA)
  if(irow=19) KRowFunction = K78R(J,M,K,L,R,JA)
  if(irow=20) KRowFunction = K66R(J,M,K,L,R,JA)
  return
121 if(irow=21) KRowFunction = K58R(J,M,K,L,R,JA)
  if(irow=22) KRowFunction = K88R(J,M,K,L,R,JA)
  if(irow=23) KRowFunction = K31R(J,M,K,L,R,JA)
  if(irow=24) KRowFunction = K90R(J,M,K,L,R,JA)
  return
125 if (irow>28 ) goto 129
  if(irow=25) KRowFunction = K39R(J,M,K,L,R,JA)
  if(irow=26) KRowFunction = K79R(J,M,K,L,R,JA)
  if(irow=27) KRowFunction = K67R(J,M,K,L,R,JA)

```

```
    if(irow=28) KRowFunction = K86R(J,M,K,L,R,JA)
    return
129 if(irow=29) KRowFunction = K69R(J,M,K,L,R,JA)
    if(irow=30) KRowFunction = K91R(J,M,K,L,R,JA)
    if(irow=31) KRowFunction = K57R(J,M,K,L,R,JA)
    if(irow=32) KRowFunction = K70R(J,M,K,L,R,JA)
    return
133     if (irow>48 ) goto 149
    if (irow>40 ) goto 141
    if (irow>36 ) goto 137
    if(irow=33) KRowFunction = K85R(J,M,K,L,R,JA)
    if(irow=34) KRowFunction = K38R(J,M,K,L,R,JA)
    if(irow=35) KRowFunction = K40R(J,M,K,L,R,JA)
    if(irow=36) KRowFunction = K80R(J,M,K,L,R,JA)
    return
137 if(irow=37) KRowFunction = K37R(J,M,K,L,R,JA)
    if(irow=38) KRowFunction = K56R(J,M,K,L,R,JA)
    if(irow=39) KRowFunction = K43R(J,M,K,L,R,JA)
    if(irow=40) KRowFunction = K59R(J,M,K,L,R,JA)
    return
141 if(irow>44) goto 145
    if(irow=41) KRowFunction = K73R(J,M,K,L,R,JA)
    if(irow=42) KRowFunction = K55R(J,M,K,L,R,JA)
    if(irow=43) KRowFunction = K81R(J,M,K,L,R,JA)
    if(irow=44) KRowFunction = K87R(J,M,K,L,R,JA)
    return
145 if(irow=45) KRowFunction = K53R(J,M,K,L,R,JA)
    if(irow=46) KRowFunction = K76R(J,M,K,L,R,JA)
    if(irow=47) KRowFunction = K65R(J,M,K,L,R,JA)
    if(irow=48) KRowFunction = K52R(J,M,K,L,R,JA)
    return
149     if (irow>56 ) goto 157
    if (irow>52 ) goto 153
    if(irow=49) KRowFunction = K82R(J,M,K,L,R,JA)
    if(irow=50) KRowFunction = K45R(J,M,K,L,R,JA)
    if(irow=51) KRowFunction = K62R(J,M,K,L,R,JA)
    if(irow=52) KRowFunction = K34R(J,M,K,L,R,JA)
    return
153 if(irow=53) KRowFunction = K48R(J,M,K,L,R,JA)
    if(irow=54) KRowFunction = K50R(J,M,K,L,R,JA)
    if(irow=55) KRowFunction = K77R(J,M,K,L,R,JA)
    if(irow=56) KRowFunction = K89R(J,M,K,L,R,JA)
    return
157     if (irow>60) goto 161
    if(irow=57) KRowFunction = K63R(J,M,K,L,R,JA)
    if(irow=58) KRowFunction = K54R(J,M,K,L,R,JA)
    if(irow=59) KRowFunction = K33R(J,M,K,L,R,JA)
    if(irow=60) KRowFunction = K71R(J,M,K,L,R,JA)
    return
161 if(irow=61) KRowFunction = K74R(J,M,K,L,R,JA)
    if(irow=62) KRowFunction = K93R(J,M,K,L,R,JA)
    if(irow=63) KRowFunction = K36R(J,M,K,L,R,JA)
    return
END
```

```

C-----
C=====
C Nice Rules, cut and pasted 7/27/93 (NOT Nice Rule list next)
  FUNCTION K92R(J,M,K,L,R, JA)
C BY ANATOL RAPOPORT
C TYPED BY AX 3/27/79 (SAME AS ROUND ONE TIT FOR TAT)
c replaced by actual code, Ax 7/27/93
c T=0
c K92R=ITFTR(J,M,K,L,T,R)
  k92r=0
  k92r = j
c test 7/30
c write(6,77) j, k92r
c77 format(' test k92r. j,k92r: ', 2i3)
  RETURN
  END
  FUNCTION K61R(ISPICK,ITURN,K,L,R, JA)
C BY DANNY C. CHAMPION
C TYPED BY JM 3/27/79
  k61r=ja ! Added 7/27/93 to report own old value
  IF (ITURN .EQ. 1) K61R = 0
  IF (ISPICK .EQ. 0) ICOOP = ICOOP + 1
  IF (ITURN .LE. 10) RETURN
  K61R = ISPICK
  IF (ITURN .LE. 25) RETURN
  K61R = 0
  COPRAT = FLOAT(ICOOP) / FLOAT(ITURN)
  IF (ISPICK .EQ. 1 .AND. COPRAT .LT. .6 .AND. R .GT. COPRAT)
+k61r = 1
  RETURN
  END
  FUNCTION K42R(JPICK,MOVEN,ISCORE,JSCORE,RANDOM, JA)
C BY OTTO BORUFSEN
C TYPED FROM FORTRAN BY AX, 1/25/79
  DIMENSION MHIST(2,2)
  k42r=ja ! Added 7/27/93 to report own old value
C INITIALIZE FIRST MOVE
  IF(MOVEN.NE.1)GOTO 20
  L3MOV=0
  L3ECH=0
  IDEF=0
  ICOOP=0
  IPICK=0
  DO 10 I=1,2
  DO 10 J=1,2
10  MHIST(I,J)=0
  GO TO 500
20  IF(MOVEN.EQ.2)GOTO 25
C UPDATE MOVE HISTORY
  MHIST(I2PCK+1,JPICK+1)=MHIST(I2PCK+1,JPICK+1)+1
25  IF(IDEF.EQ.0)GOTO 30
C OPPONENT HAS BEEN PROVED "RANDOM" OR
C "DEFECTIVE",I DEFECT FOR 25 MOVES
  K42R=1
  GO TO 100
30  IF(IPICK.EQ.0.OR.JPICK.EQ.0)GOTO 40
C MUTUAL DEFECTIONS ON LAST MOVE.

```

```

        L3MOV=L3MOV+1
        IF(L3MOV.LT.3)GOTO 50
C MUTUAL DEFECTION ON
C LAST THREE MOVES.
C I COOPERATE ONCE ON NEXT MOVE.
        K42R=0
        L3MOV=0
        L3ECH=0
        GOTO 100
C ONE (OR BOTH) COOPERATED ON LAST MOVE.
40      L3MOV=0
        IF(IPICK.EQ.JPICK)GOTO 45
        IF(JPICK.NE.I2PCK.OR.IPICK.NE.J2PCK)GOTO 45
C ECHO-EFFECT ON LAST MOVE.
        L3ECH=L3ECH+1
        IF(L3ECH.LT.3)GOTO 50
C ECHO-EFFECT ON LAST THREE MOVES.
C MY NEXT DEFECTION WILL BE SUBSTITUTED
C BY A COOPERATION.
        L3ECH=0
        L3MOV=0
        ICOOP=1
        GOTO 50
45      L3ECH=0
C PLAY 'TIT FOR TAT' AS MAIN RULE.
50      K42R=JPICK
100     IF(MOD(MOVEN-2,25).NE.0.OR.MOVEN.EQ.2)GOTO 650
C ON EVERY 25 MOVES:
C CHECK IF OPPONENT SEEMS TO BE
C 'RANDOM' OR 'DEFECTIVE'.
        IDEF=0
        JNCOP=MHIST(1,1)+MHIST(2,1)
C IS OPPONENT 'RANDOM'?
        IF(JNCOP.GT.17)GOTO 155
        IF(JNCOP.LT.8)GOTO 130
        IF(100*MHIST(1,1)/JNCOP.LT.70)IDEF=1
        GOTO 155
C IS OPPONENT 'DEFECTIVE'?
130     IF(JNCOP.LT.3)IDEF=1
155     DO 160 I=1,2
        DO 160 J=1,2
160     MHIST(I,J)=0
        IF(IDEF.EQ.0)GOTO 650
C OPPONENT SEEMS TO BE
C 'RANDOM' OR 'DEFECTIVE'.
C I DEFECT FOR NEXT 25 MOVES.
        ICOOP=0
        L3MOV=0
        L3ECH=0
        GOTO 600
C I COOPERATE.
500     K42R=0
        GOTO 650
C I DEFECT.
600     K42R=1
650     IF(ICOOP.EQ.0.OR.K42R.EQ.0)GOTO 660
        ICOOP=0

```

```

        K42R=0
660   I2PCK=IPICK
      J2PCK=JPICK
      IPICK=K42R
      RETURN
      END
      FUNCTION K49R(J,M,K,L,R, JA)
C BY ROB CAVE
C TYPED BY JM
      k49r=ja      ! Added 7/27/93 to report own old value
      IF (M .EQ. 1) JDSUM = 0
C JDSUM IS THE TOTAL NUMBER OF DEFECTIONS SO FAR
      IF (J .EQ. 1) JDSUM = JDSUM + 1
      JDPC = (100 * JDSUM) / M
C JDPC IS THE PERCENTAGE OF DEFECTIONS SO FAR
      IF (J .EQ. 0) K49R = 0
      IF ((J .EQ. 1) .AND. (JDSUM .LE. 17)) K49R = INT(R + .5)
      IF ((J .EQ. 1) .AND. (JDSUM .GT. 17)) K49R = 1
C IF OPONENT IS OVERLU DEFECTIVE OR APPEARS
C TO BE RANDOM, THEN GIVE UP
      IF ((M .GT. 19) .AND. (JDPC .GT. 79)) K49R = 1
      IF ((M .GT. 29) .AND. (JDPC .GT. 65)) K49R = 1
      IF ((M .GT. 39) .AND. (JDPC .GT. 39)) K49R = 1
      RETURN
      END
      FUNCTION K44R(J,M,K,L,R, JA)
C BY WM. ADAMS
C EDITED FROM BASIC BY AX, 1/26/79
      k44r=ja      ! Added 7/27/93 to report own old value
      IF(M.NE.1) GOTO 520
C COUNT HIS DEFECTS
      MC=0
C ADJUST FACTOR
      F=2
C NR. DEFECTS ALLOWED
      AM=4
C COOP AT FIRST
520   IF(M.LT.3) GOTO 1800
      MC=MC+J
C COOP UNTIL THRESHOLD
      IF(MC.LT.AM) GOTO 1800
      IF(MC.EQ.AM) GOTO 1900
C ADJUST: LOWER THRESHOLD
      AM=AM/F
      MC=0
C ANOTHER CHANCE WITH PROB. P
      IF(R.LT.AM) GOTO 1800
1900  K44R=1
      RETURN
1800  K44R=0
      RETURN
      END
      FUNCTION K60R(J,M,K,L,R, JA)
C BY JIM GRAASKAMP AND KEN KATZEN
C FROM CARDS BY JM 2/22/79
      k60r=ja      ! Added 7/27/93 to report own old value
      IF (M-1)1,1,2

```



```

1      ID=0
      K60R=0
      GO TO 50
2      IF (ID-1)3,4,4
3      K60R=J
      IF (M-11)50,5,6
5      IF (K-23)51,50,50
6      IF (M-21)50,7,8
7      IF(K-53)51,50,50
8      IF (M-31)50,9,10
9      IF (K-83)51,50,50
10     IF (M-41)50,11,12
11     IF (K-113)51,50,50
12     IF (M-51)50,13,14
13     IF (K-143)51,50,50
14     IF (M-101)50,15,50
15     IF (K-293)51,50,50
51     ID=1
4      K60R=1
50     RETURN
      END
      FUNCTION K41R(J,M,K,L,R, JA)
C BY HERB WEINER
C TYPED BY AX, 1/29/79
      INTEGER LAST(12)
      k41r=ja      ! Added 7/27/93 to report own old value
      IF(M.NE.1)GOTO 10
      ICASE=1
      IFORGV=0
      DO 5 I=1,12
5      LAST(I)=0
10     GOTO(100,200,300),ICASE
100    K41R=J
      ICASE=J+1
      GOTO 400
200    K41R=J
      ICASE=3
      IF(J.EQ.1)ICASE=1
      GOTO 400
300    K41R=J
      IF(IFORGV.LT.M)K41R=0
      IFORGV=IFORGV+20*J
      ICASE=1
400    LSUM=LAST(1)
      DO 405 I=2,12
      LSUM=LSUM+LAST(I)
405    LAST(I-1)=LAST(I)
      LAST(12)=J
      IF(LSUM.GE.5)K41R=1
      RETURN
      END
      FUNCTION K84R(JP,M,IS,JS,R, JA)
C BY T NICOLAUS TIDEMAN AND PAULA CHIERUZZI
C TYPED BY JM 3/31/79
      k84r=ja      ! Added 7/27/93 to report own old value
      K84R = 1
      IF (M .GT. 1) GOTO 2

```

```

        ISIG = 0
        DS = 0
        JQ = 0
        FJD = 0
        JDR = 0
        FM = 0
        GOTO 3
2       IF (JP .EQ. 1) FJD = FJD + 1
        IF (ISIG .EQ. 1) GOTO 5
        FM = M
        IF (JQ .EQ. 0 .AND. JP .EQ. 1) JDR = JDR + 1
3       IF (IS - JS - DS - 5 * JDR * (JDR - 1) / 2 .GE. 0)
1       K84R = 0
        IF (K84R .EQ. 1) GOTO 4
1       JQ = JP
        RETURN
4       IF ((JQ - JP) .LT. 0 .OR. (M - ISIG) .LT. 10) GOTO 1
        IF (ABS(FJD - (FM - 1.) / 2.) .LT. (1.5 * SQRT(FM - 1.)
1 )) GOTO 1
        ISIG = 1
        JQ = JP
        GOTO 6
5       ISIG = M
        JQ = 0
        JDR = 0
        DS = IS - JS
6       K84R = 0
        RETURN
        END
        FUNCTION K32R(J,M,K,L,R, JA)
C BY CHARLES KLUEPFEL
C EDITED FROM BASIC BY AX, 1.19.79
        k32r=ja      ! Added 7/27/93 to report own old value
        IF(M.GT.1) GO TO 520
C # OF HIS COOPS AFTER MY DEF.
        C1=0
C # OF HIS DEFECTIONS AFTER MY DEF.
        C2=0
C # OF HIS COOPS AFTER MY COOPERATION
        C3=0
C # OF HIS DEF. AFTER MY COOPERATION
        C4=0
C HIS 3RD PREV. CHOICE
        J2=0
C HIS 2ND PREV. CHOICE
        J1=0
C MY 2ND PREV. CHOICE
        I2=0
C MY PREV. CHOICE
        I1=0
C PROB. OF MY RESPONDING IN KIND
        P=0
        GO TO 550
520 IF(M.EQ.2) GO TO 550
C RECORD HIS RESPONSES
        IF(I2.EQ.0) GO TO 530
        IF(J.EQ.0) GO TO 528

```

```

        C2=C2+1
        GO TO 540
528 C1=C1+1
        GO TO 540
530 IF(J.EQ.0) GO TO 537
        C4=C4+1
        GO TO 540
537 C3=C3+1
C AFTER M. 26 TRY DETECTING RNDOM
540 IF(M.LT.27) GO TO 550
        IF (C1.LT.((C1+C2) - 1.5*SQRT(C1+C2)) / 2 ) GO TO 550
        IF (C4.LT.((C3+C4) - 1.5*SQRT(C3+C4)) / 2 ) GO TO 550
        K32R=1
        GO TO 590
550 K32R=0
        IF(J1.NE.J) GO TO 570
        IF(J2.NE.J1) GO TO 580
C RESPOND IN KIND TO 3 IN ROW.
        K32R = J
        GO TO 590
C PROB .6 OF GIVING DEF. AFTER SINGLE DEF.
570 P=.6
        IF(J.EQ.1) GO TO 585
C PROB .7 GIVING COOP AFTER SINGLE COOP.
        P=.7
        GO TO 585
C PROB .9 RESP. IN KIND TO 2 IN ROW
580 P=.9
585 K32R=J
        IF(R.LT.P) GO TO 590
        K32R=1-J
C PUSH BACK CHOICES
590 J2=J1
        J1=J
        I2=I1
        I1=K32R
        RETURN
        END
        FUNCTION K35R(J,M,K,L,R, JA)
C BY ABRAHAM GETZLER
C TYPED FROM FORTRAN BY AX, 1,17,79
        k35r=ja      ! Added 7/27/93 to report own old value
        IF(M.EQ.1) FLACK=0.
C FLACK IS THE RELATIVE RECENT UNTRUSTWORTHINESS OF MY PPOONENT
        FLACK = (FLACK + J) * .5
C DEFLECTIONS HAVE A HALF-LIFE OF ONE ROUND
        K35R = 0
        IF (FLACK.GT.R) K35R=1
        RETURN
        END
        FUNCTION K68R(J,M,K,L,R, JA)
C BY FRANSOIS LEYVRAZ
C EDITED FROM BASIC BY AX, 3/10/79
C TYPED BY JM 3/16/79
        k68r=ja      ! Added 7/27/93 to report own old value
        IF (M .EQ. 1) GOTO 600
        IF (J1 * J .EQ. 1) GOTO 540

```

```

        IF (J2 * 2 + J1 + J * 2 + J .EQ. 1) GOTO 550
        IF (J2 * 2 + J1 * 2 + J .EQ. 1) GOTO 560
        K68R = 0
        GOTO 650
540    IF (R .LT. 0.75) GOTO 550
        K68R = 0
        GOTO 650
550    K68R = 1
        GOTO 650
560    IF (R .LT. 0.5) GOTO 550
        K68R = 0
        GOTO 650
600    J2 = 0
        J1 = 0
        K68R = 0
        RETURN
650    J2 = J1
        J1 = J
        RETURN
END
        FUNCTION K72R(J,M,K,L,R, JA)
C BY EDWARD C WHITE, JR.
C TYPED BY JM 3/22/79; COR BY AX 3/31/79
        k72r=ja      ! Added 7/27/93 to report own old value
        IF (M .EQ. 1) JOLD = 0
        K72R = 0
        IF (M .EQ. 1) JCOUNT = 0
        JOLD = J
        IF (JOLD .EQ. 1) JCOUNT = JCOUNT + 1
        N = 1
        IF (JOLD .EQ. 1 .AND. M .GT. 10) N = ALOG(FLOAT(M))
        IF (R .LE. ((N * JCOUNT) / M)) K72R = 1
        RETURN
END
        FUNCTION K46R(J,M,K,L,R, JA)
C BY GRAHAM J. EATHERLEY
C TYPED FROM FORTRAN BY AX, 1/26/79
        k46r=ja      ! Added 7/27/93 to report own old value
        IF(M.EQ.1) NJ=0
        NJ=NJ+J
        K46R=0
        IF(J.EQ.0) RETURN
        P=FLOAT(NJ)/FLOAT(M-1)
        IF(R.LT.P) K46R=1
        RETURN
END
        FUNCTION K83R(JPICK,MOVEN,I,J,RAND, JA)
C BY PAUL E BLACK
C TYPED BY JM 3/31/79
        DIMENSION JHIS(5)
        k83r=ja      ! Added 7/27/93 to report own old value
        IF (MOVEN .GT. 5) GOTO 20
        IF (MOVEN .NE. 1) GOTO 10
        JTOT = 0
        MCNT = 1
10     K83R = 0
        JHIS(MOVEN) = JPICK

```

```

JTOT = JTOT + JPICK
RETURN
20 JTOT = JTOT - JHIS(MCNT) + JPICK
JHIS(MCNT) = JPICK
MCNT = MCNT + 1
IF (MCNT .GT. 5) MCNT = 1
K83R = 0
IF (RAND * 25 .LT. JTOT * JTOT - 1) K83R = 1
RETURN
END
FUNCTION K64R(J,M,K,L,R, JA)
C BY BRIAN YAMACHI
C EDITED FROM BASIC BY AX, 2/28/79
C TYPED BY JM 3/1/79
IMPLICIT INTEGER (A-Z)
REAL R
DIMENSION A(2,2)
k64r=ja ! Added 7/27/93 to report own old value
IF (M .GT. 1) GOTO 640
E = 0
F = 0
DO 560 C = 1,2
DO 560 D = 1,2
560 A(C,D) = 0
X = 1
Y = 1
K64R = 0
Y = K64R + 1
RETURN
640 IF (A(X,Y) .GE. 0) K64R = 0
IF (A(X,Y) .LT. 0) K64R = 1
IF (J .EQ. 0) A(X,Y) = A(X,Y) + 1
IF (J .EQ. 1) A(X,Y) = A(X,Y) - 1
X = J + 1
Y = K64R + 1
IF (J .EQ. 0) E = E + 1
IF (J .EQ. 1) F = F + 1
P = E - F
IF (P .LT. 0) P = -P
IF ((M .GT. 40) .AND. (10 * P .LT. M)) K64R = 1
RETURN
END
FUNCTION K66R(J,M,K,L,R, JA)
C BY RAY MIKKELSON
C TYPED BY JM 3/16/80
k66r=ja ! Added 7/27/93 to report own old value
IF (M .GT. 1) GOTO 20
D = 0
J2 = -3
20 D = D + J
RR = D / FLOAT(M)
J2 = J2 - 1 + 3 * J
IF (J2 .GT. 10) J2 = 10
IF (J2 .LT. -5) J2 = -5
IF (M .LT. 3) GOTO 90
IF (J2 .LT. 3) GOTO 90
IF (M .GT. 10) GOTO 58

```

```

        J2 = -1
        GOTO 80
58     IF (RR .LT. .15) GOTO 90
80     K66R = 1
        GOTO 95
90     K66R = 0
95     RETURN
      END
      FUNCTION K58R(J,M,K,L,R, JA)
C BY GLEN ROWSAM
C TYPED BY JM
      k58r=ja      ! Added 7/27/93 to report own old value
      IF (M .GT. 1) GOTO 99
      KAM = 0
      NPHA = 0
99     IF (KAM .GT. 6) GOTO 87
      IF (NPHA .GE. 1) GOTO 89
      IF ((M / 18) * 18 .EQ. M .AND. KAM .GT. 2) KAM = KAM - 1
      IF ((M / 6) * 6 .NE. M) GOTO 88
      IF (K .LT. M) GOTO 10
      IF (K * 10 .LT. M * 15) GOTO 11
      IF (K .LT. M * 2) GOTO 12
      IF (K * 10 .LT. M * 25) GOTO 13
      GOTO 88
10     KAM = KAM + 2
11     KAM = KAM + 1
12     KAM = KAM + 1
13     KAM = KAM + 1
      NPHA = 2
      GOTO 87
89     NPHA = NPHA - 1
      IF (NPHA .EQ. 0) GOTO 87
88     K58R = 0
      GOTO 86
87     K58R = 1
86     RETURN
      END
      FUNCTION K88R(J,M,K,L,R, JA)
C BY SCOTT APPOLD
C EDITED FROM NEAR-FORTRAN BY AX 3/27/79
C TYPED BY JM 3/31/79
      k88r=ja      ! Added 7/27/93 to report own old value
      K88R = 0
      IF (M .NE. 1) GOTO 10
      MMC = 0
      LMV = 0
      MP = 0
      MMV = 0
      MP2 = 0
      MMD = 1
      DFLG = 0
10     IF (M .LT. 2) GOTO 20
      IF (MMV .NE. 0) GOTO 15
      MMC = MMC + 1
      MP = MP + J
      PRC = FLOAT(MP) / FLOAT(MMC)
      GOTO 20

```

```

15   MMD = MMD + 1
      MP2 = MP2 + J
      PRD = FLOAT(MP2) / FLOAT(MMD)
20   CONTINUE
      IF (M .GT. 4) GOTO 25
      K88R = 0
      GOTO 30
25   IF (.NOT.(J .EQ. 1 .AND. DFLG .EQ. 0)) GOTO 28
      DFLG = 1
      K88R = 0
      GOTO 30
28   IF (MMV .EQ. 0 .AND. R .LT. PRC) K88R = 1
      IF (MMV .EQ. 1 .AND. R .LT. PRD) K88R = 1
30   CONTINUE
      MMV = LMV
      LMV = K88R
      RETURN
      END
      FUNCTION K31R(J,M,K,L,R, JA)
C BY PAULA GAIL GRISELL
C   EDITED FROM BASIC BY AX, 1.17.79
      k31r=ja      ! Added 7/27/93 to report own old value
      IF(M.EQ.1) S=0.
      S=S+J
      A=S/M
      K31R=1
      IF (A .LT..5) K31R=0
      RETURN
      END
      FUNCTION K90R(J,M,K,L,R, JA)
C BY JOHN MAYNARD SMITH
C   TYPED BY AX 3/27/79 (SAME AS ROUND ONE TIT FOR TWO TATS)
      k90r=ja      ! Added 7/27/93 to report own old value
C   recoded by Ax 7/27/93
      if(m.eq.1) jold=0
      k90r=0
      if((jold.eq.1).and.(j.eq.1)) k90r=1
      jold=j
      RETURN
      END
      FUNCTION K79R(J,M,K,L,R, JA)
C BY DENNIS AMBUEHL AND KEVIN HICKEY
C   FROM CARDS BY JM 3/16/79
      DIMENSION JBACK(5)
C   COOPERATES IF OPPONENT COOPERATED ON MAJORITY OF LAST PLAYS
      k79r=ja      ! Added 7/27/93 to report own old value
      IF (M.EQ.1) GO TO 3000
      IF (M.LT.6) GO TO 4000
      I1 = 0
      DO 1500 I2 = 1,5
1500  I1 = I1 + JBACK(I2)
      IF (I1.LT.3) GO TO 1000
      K79R = 1
      GO TO 2000
3000  DO 2500 I2 = 1,5
2500  JBACK(I2) = 0
1000  K79R = 0

```

```

2000 DO 3500 I2 = 1,4
3500 JBACK(I2) = JBACK(I2 + 1)
      JBACK(5) = J
      RETURN
4000 K79R = J
      GO TO 2000
      END
      FUNCTION K86R(JPICK,MOVEN,ISCORE,JSCORE,RANDOM, JA)
C BY BERNARD GROFMAN
C FROM CARDS BY JM 3/27/79
      DIMENSION IOPPNT(999)
      k86r=ja      ! Added 7/27/93 to report own old value
      IOPPNT(MOVEN) = JPICK
      MYOLD = K86R
      IF (MOVEN .GT. 2) GOTO 10
      K86R = 0
      RETURN
10     IF (MOVEN. GT. 7) GOTO 20
      K86R = JPICK
      RETURN
20     IPREV7 = 0
      J = MOVEN - 7
      K = MOVEN - 1
      DO 25 I = J,K
25     IPREV7 = IPREV7 + IOPPNT(I)
      IF (MYOLD .EQ. 0 .AND. IPREV7 .LE. 2) K86R = 0
      IF (MYOLD .EQ. 0 .AND. IPREV7 .GT. 2) K86R = 1
      IF (MYOLD .EQ. 1 .AND. IPREV7 .LE. 1) K86R = 0
      IF (MYOLD .EQ. 1 .AND. IPREV7 .GT. 1) K86R = 1
      RETURN
      END
      FUNCTION K91R(J,M,K,L,R, JA)
C BY JONATHAN PINKLEY
C MODIFIED FROM K15C BY JM 3/27/79
      DIMENSION IPOL(11,4), QC(4), QN(4), E(11)
      k91r=ja      ! Added 7/27/93 to report own old value
      IF (M .NE. 1) GO TO 30
C INITIAL BELIEFS
      X = .999
      PX = .001
      Y = .001
      PY = .999
      Z = .999
      PZ = .001
      W = .001
      PW = .999
      QC(1) = 1.999
      QC(2) =1.999
      QC(3) = 0.001
      QC(4) = 0.001
      DO 10 N = 1, 4
10     QN(N) = 2
C DEFINE POLICIES(FIRST,WHAT IF OUTCOME=1)
      DATA IPOL /4*0, 7*1, 0, 3*1, 3*0, 4*1, 3*0, 1, 2*0, 1, 0, 0, 1, 1,
1 2*0, 1, 0, 0, 1, 0, 0, 1, 0, 1/
      IOLD=0
      K91R = 0

```



```

N = 0
GO TO 100
C UPDATE STATS OF HIS CONTINGENCIES
C N IS OUTCOME OF M-2
  30 IF (M .LE. 2) GO TO 100
    IF (J .EQ. 0) QC(N) = QC(N) + 1
    QN(N) = QN(N) + 1
C REVERSE Y AND Z
  GO TO (40, 60, 50, 70), N
  40 X = QC(1) / QN(1)
    PX = 1 - X
    GO TO 100
  50 Y = QC(3) / QN(3)
    PY = 1 - Y
    GO TO 100
  60 Z = QC(2) / QN(2)
    PZ = 1 - Z
    GO TO 100
  70 W = QC(4) / QN(4)
    GO TO 100
C CALC EXPECTATIONS OF POLICIES
100 E(1) = (3*Z) / (Z + PX)
    E(2) = (3*(Y*Z + W*PZ) + 5*Z*PX + PX*PZ) / (Y*Z + W*PZ + PX + Z*
1 PX + PX*PZ)
    E(3) = (3*W*Y + 5*W*PX + PX*PZ) / (W*Y + 2*W*PX + PX*PZ)
    E(4) = (3*W*PY + 5*Z*PX + PX*PY) / (W*PY + PX*PY + Z*PX + PX*PY)
    E(5) = (3*Z + 5*X*Z + Z*PX) / (1 - X*Y - W*PX + 2*Z)
    E(6) = (8*W*Z + Z*PX) / (2*W*Z + W*PY + Z*PX)
    E(7) = (3*Z*PY + 5*X*Z + Z*PY) / (2*Z*PY + PW*PY + X*Z)
    E(8) = (3*(Y*Z + W*PZ) + 5*(Z*PW + W*X) + 1 - X*Y - Z*PY) / (Y *Z +
1 W*PZ + 2 - 2*X*Y - W*PX + Z*PW + W*X - Z*PY)
    E(9) = (3*W*Y + 5*W + 1 - X*Y - Z*PY) / (2*W + 1 - X*Y - Z*PY)
    E(10) = (3*W*PY + 5*(Z*PW + W*X) + PY) / (PY + Z*PW + W*X + PY)
    E(11) = (5*W + PY) / (W + PY)
C FIND POL WITH MAX E
  IBEST = 1
  BESTE = E(1)
  DO 80 I = 2, 11
    IF (E(I) .LE. BESTE) GO TO 80
    IBEST = I
    BESTE = E(I)
  80 CONTINUE
C CALC OUTCOME FOR USE IN CHOICE AND NEXT MV STATS
  90 N = 2 * IOLD + J + 1
C CHOICE(CHOSEN POLICY,PREV OUTCOME)
  K91R = IPOL(IBEST,N)
  IOLD=K91R
  RETURN
  END
  FUNCTION K57R(J,M,K,L,R, JA)
C BY RUDY NYDEGGER
C TYPED BY AX, 3/27/79 (SAME AS ROUND ONE NYDEGR)c Replaced by Nydegr retyped from rnd 1 by
Ax, 7/27/93
c T=0
c K57R=NYDEGR(J,M,K,L,T,R)
c RETURN
c END

```

```

k57r=ja      ! Added 7/27/93 to report own old value
if(m.ne.1) goto 5
k57r = 0
n = 0
c update 3 move history
5  N = 4 * (n-16*(N/16)) + 2 * k57r + J
if(m.gt.3) goto 8
k57r=j
if(m.eq.3 .and. n.eq.6) k57r=1
return
c coop if 0, 27, 28, 32, 40-4, 46-8, 56-7,59-60,62-63
8  k57r=1
   if(n-39) 10,110,50
10 if(n) 100,100,20
20 if(n-28) 30,100,40
30 if(n-27) 110,100,100
40 if(n-32) 110,100,110
50 if(n-45) 100,110,60
60 if(n-49) 100,110,70
70 if(n-58) 80,110,90
80 if(n-55) 110,110,100
90 if(n-61) 100,110,100
100 k57r = 0
110 return
end
FUNCTION K70R(J,M,K,L,R, JA)
C BY ROBERT PEBLY
C EDITED FROM BASIC BY AX 3/10/79
C TYPED BY JM 3/16/79
k70r=ja      ! Added 7/27/93 to report own old value
IF (M .EQ. 1) JZ = 0
IF (JZ .EQ. J) GOTO 510
K70R = 0
IF (R .GT. .2) K70R = 1
506 JZ = K70R
RETURN
510 K70R = JZ
GOTO 506
END
FUNCTION K85R(J,M,K,L,R, JA)
C BY ROGER B FALK AND JAMES M LANGSTED
C EDITED FROM BASIC BY AX 3/18/79
C TYPED BY JM 4/4/79
C INITIALIZE ON FIRST MOVE AND COOPERATE
IMPLICIT REAL (A-Z)
INTEGER J,M,K,L,K85R
k85r=ja      ! Added 7/27/93 to report own old value
IF (M .NE. 1) GOTO 100
J2 = 0
J4 = 0
J8 = 0
J0 = 0
F4 = 0
F8 = 0
F0 = 0
K85R = 0
F1 = 0

```

```

C = 0
D = 0
T = 0
I1 = 0
I2 = 0
I3 = 0
I4 = 0
GOTO 900
C SERVICE SHIFT REGESTERS J0 AND F0
100  J5 = J0 / 1E07
      J3 = INT(J5)
      J8 = J5 - J3
      J8 = J8 * 1E07
      F5 = F0 / 1E07
      F3 = INT(F5)
      F8 = F5 - F3
      F8 = F8 * 1E07
      J0 = J8 * 10 + 5
      F0 = F8 * 10 + 5
C SERVICE COUNTERS TO TALLY NUMBER OF TIMES
C HIS VARIOUS RESPONSES FOLLOW MY VARIOUS RESPONSES
      IF (F1 .EQ. 0) GOTO 175
      IF (J .EQ. 0) I1 = I1 + 1
      IF (J .EQ. 1) I2 = I2 + 1
      GOTO 185
175  IF (J .EQ. 0) I3 = I3 + 1
      IF (J .EQ. 1) I4 = I4 + 1
C CHECK FOR RANDOMNESS AFTER FIRST 20 MOVES
185  IF (M .LE. 20) GOTO 245
      I5 = I1 + 1E-6
      I6 = I2 + 1E-6
      X8 = I3 + 1E-6
      I8 = I4 + 1E-6
      A = I5 / I6
      B = X8 / I8
      IF (A .GT. 1.5) GOTO 245
      IF (A .LT. .5) GOTO 245
      IF (B .GT. 1.5) GOTO 245
      IF (B .LT. .5) GOTO 245
      GOTO 910
C CHECK IF WE ARE IN TIT FOR TAT MODE
245  IF (T .EQ. 1) GOTO 920
C CHECK IF HE CONTINUALY DEFECTS
      IF (J0 .EQ. 11111111)GOTO 920
C CHECK IF WE ARE IN D THEN C MODE
      IF (C .EQ. 1) GOTO 980
C CHECK IF HE HAS COOPERATED TWICE IN A ROW
C IN FIRST 30 MOVES
      Z1 = J0 / 100
      Z2 = INT(Z1)
      Z3 = Z1 - Z2
      J2 = Z3 * 100
      IF (M .GT. 30) GOTO 295
      IF (J2 .NE. 11) GOTO 295
      GOTO 390
C CHECK IF HE IS PLAYING TIT FOR TAT
295  Z4 = J0 / 10000

```

```

Z5 = INT (Z4)
Z6 = Z4 - Z5
J4 = Z6 * 10000
W8 = F0 / 10000
Z8 = INT(W8)
Z9 = W8 - Z8
F4 = Z9 * 10000
IF (J4 .NE. 1011) GOTO 350
IF (F4 .NE. 111) GOTO 350
GOTO 930
C CHECK IF HE IS OVER 3 DEFECTS AHEAD
350 Y1 = I2 + I4
Y2 = I1 + I2 + 3
IF (Y1 .GE. Y2) GOTO 910
C USE BASIC RULES
IF (K85R .NE. 1) GOTO 380
IF (J .NE. 0) GOTO 380
GOTO 940
380 IF (D .EQ. 1) GOTO 995
IF (K85R .NE.0) GOTO 400
390 IF (J .NE. 0) GOTO 400
GOTO 900
400 IF (K85R .NE. 0) GOTO 415
IF (J .NE. 1) GOTO 415
GOTO 910
415 IF (K85R .NE. 0) GOTO 950
C COOPERATE RETURN
900 F1 = K85R
K85R = 0
RETURN
C DEFECT RETURN
910 F1 = K85R
K85R = 1
D = 0
RETURN
C TIT FOR TAT MODE RETURN
920 T = 1
K85R = J
RETURN
C CC RETURN (FIRST TIME)
930 C = 1
GOTO 981
C DEFECT AND RESET D RETURN
940 F1 = K85R
K85R = 1
D = 0
RETURN
C D THEN C RETURN (FIRST TIME)
950 F1 = K85R
K85R = 1
D = 1
RETURN
C CC RETURN (SECOND TIME)
980 C = 0
981 F1 = K85R
K85R = 0
RETURN

```

```

C D THEN C RETURN (SECOND TIME)
995  F1 = K85R
      K85R = 0
      D = 0
      RETURN
      END
      FUNCTION K38R(J,M,K,L,R, JA)
C BY NELSON WEIDERMAN
C TYPED BY AX FROM FORTRAN, 1.17.79
C
C RULE: DEFECT FOREVER AFTER THREE CONSECUTIVE
C DEFECTIONS BY OPPONENT
C JHIS STORES LAST THREE OPPONENT MOVES AS 4*J3 + 2*J2 +J1
C WHERE J1 IS MOST RECENT MOVE AND J3 IS LEAST RECENT
      k38r=ja      ! Added 7/27/93 to report own old value
      IF(M.NE.1) GO TO 10
      MOVE=0
      JHIS=0
10  CONTINUE
      IF(MOVE.EQ.1) GO TO 20
      IF(JHIS.GE.4)JHIS=JHIS-4
      JHIS=JHIS*2
      JHIS=JHIS+J
      IF(JHIS.EQ.7)MOVE=1
20  CONTINUE
      K38R=MOVE
      RETURN
      END
      FUNCTION K40R(J,M,K,L,R, JA)
C BY ROBERT ADAMS
C EDITED FROM BASIC BY AX, 1,18,79
      k40r=ja      ! Added 7/27/93 to report own old value
      IF(M.NE.1) GO TO 505
      S=3
      W=0
      Q=.8
505 S=S+1
      IF(J.NE.1) GO TO 510
      W=W+1
      Q=Q/2
510 IF(M.GE.3) GO TO 520
      K40R=0
      RETURN
520 IF(J.EQ.1) GO TO 522
      GO TO 530
522 W=W+1
      IF(W.GT.2.AND.(W/3.EQ.IFIX(W/3)).OR.(W-1)/3.EQ.IFIX((W-1)/3)) GO TO 901
      GO TO 550
901 S=1
      Q=Q/2
      GO TO 580
530 GO TO 580
550 IF(R.GE.Q) GO TO 560
      K40R=0
      Q=Q/2
      RETURN
560 Q=Q/2

```

```

      K40R=1
      RETURN
580 IF(S.EQ.1.OR.S.EQ.2) GO TO 1000
c Ax added ( )
590 IF(W.GT.2.AND.(W/3.EQ.IFIX(W/3).OR.(W-1)/3.EQ.IFIX((W-1)/3)))
      1GO TO 901
      K40R=0
      RETURN
1000 K40R=1
      RETURN
END

```

```

      FUNCTION K80R(J,M,K,L,R, JA)
C BY ROBYN M DAWES AND MARK BATELL
C TYPED BY JM 3/22/79
      k80r=ja      ! Added 7/27/93 to report own old value
      IF (M .EQ. 1) GOTO 10
      IF (MODE .EQ. 1) GOTO 35
      IF (J .EQ. 1) GOTO 20
      GOTO 15
5      INOC = M - INOD
      T1 = 1.6667 ** INOD
      T2 = 0.882 ** INOC
      TEST = T1 * T2
      IF (TEST .GE. 5.) GOTO 30
      GOTO 15
10     MODE = 0
      INOD = 0
      INOC = 0
      T1 = 0
      T2 = 0.
      TEST = 0.
15     K80R = 0
      GOTO 40
20     INOD = INOD + 1
      GOTO 5
30     MODE = 1
35     K80R = 1
40     RETURN
END

```

```

      FUNCTION K37R(J,M,K,L,R, JA)
C BY GEORGE LEFEVRE
C EDITED FROM BASIC BY AX, 2/11/79
C TYPED BY JM
      k37r=ja      ! Added 7/27/93 to report own old value
      IF (M .GT. 1) GOTO 500
      ND = 0
500    K37R = 0
C TIMES HE DEFECTED
      ND = ND + J
      IF (5*ND .GT. M) K37R = 1
      RETURN
END

```

```

      FUNCTION K56R(LASTMV,MOVEN,K,L,R, JA)
C THIS ALGORITHM IS EXACTLY THE REVISED DOWNING METHID.
C BY STANLEY F QUAYLE
C TYPED BY JM
      INTEGER LASTMV,MOVEN

```

```

    INTEGER PAST,NICE1,NICE2
    REAL GOOD,BAD,C,ALT
    INTEGER DEFECT, COOP
    INTEGER TOTCOP,TOTDEF
k56r=ja      ! Added 7/27/93 to report own old value
    IF (MOVEN - 2) 1,4,2
1    GOOD = 1.0
    BAD = 0.0
    PAST = 0
    TOTCOP = 0
    TOTDEF = 0
    NICE1 = 0
    NICE2 = 0
    COOP = 0
    DEFECT = 1
    GOTO 4
2    IF (PAST .EQ. DEFECT) GOTO 3
    IF (LASTMV .EQ. COOP) NICE1 = NICE1 + 1
    TOTCOP = TOTCOP + 1
    GOOD = FLOAT(NICE1) / FLOAT(TOTCOP)
    GOTO 4
3    IF (LASTMV .EQ. COOP) NICE2 = NICE2 + 1
    TOTDEF = TOTDEF + 1
    BAD = FLOAT(NICE2) / FLOAT(TOTDEF)
4    PAST = K56R
    C = 6.0 * GOOD - 8.0 * BAD - 2.0
    ALT = 4.0 * GOOD - 5.0 * BAD - 1.0
    IF (C .GE. 0.0 .AND. C .GE. ALT) GOTO 5
    IF (C .GE. 0.0 .AND. C .LT. ALT) GOTO 6
    IF (ALT .GE. 0.0) GOTO 6
    K56R = DEFECT
    GOTO 7
5    K56R = COOP
    GOTO 7
6    K56R = 1 - K56R
7    RETURN
    END

```

```

    FUNCTION K59R(LASTMV,MOVEN,K,L,R, JA)
C BY LESLIE DOWNING
C TYPED BY AX, 3/27/79 (SAME AS ROUND ONE REV.DOWNING)
c Redone as copy of K56=RevDowning by Ax, 7/27/93
c    INTEGER XDOWNC
c    T=0
c    K59R=XDOWNC(J,M,K,L,T,R)
c    RETURN
c    END
    INTEGER LASTMV,MOVEN
    INTEGER PAST,NICE1,NICE2
    REAL GOOD,BAD,C,ALT
    INTEGER DEFECT, COOP
    INTEGER TOTCOP,TOTDEF
k59r=ja      ! Added 7/27/93 to report own old value
    IF (MOVEN - 2) 1,4,2
1    GOOD = 1.0
    BAD = 0.0
    PAST = 0

```

```

TOTCOP = 0
TOTDEF = 0
NICE1 = 0
NICE2 = 0
COOP = 0
DEFECT = 1
GOTO 4
2 IF (PAST .EQ. DEFECT) GOTO 3
  IF (LASTMV .EQ. COOP) NICE1 = NICE1 + 1
  TOTCOP = TOTCOP + 1
  GOOD = FLOAT(NICE1) / FLOAT(TOTCOP)
  GOTO 4
3 IF (LASTMV .EQ. COOP) NICE2 = NICE2 + 1
  TOTDEF = TOTDEF + 1
  BAD = FLOAT(NICE2) / FLOAT(TOTDEF)
4 PAST = K59R
  C = 6.0 * GOOD - 8.0 * BAD - 2.0
  ALT = 4.0 * GOOD - 5.0 * BAD - 1.0
  IF (C .GE. 0.0 .AND. C .GE. ALT) GOTO 5
  IF (C .GE. 0.0 .AND. C .LT. ALT) GOTO 6
  IF (ALT .GE. 0.0) GOTO 6
  K59R = DEFECT
  GOTO 7
5 K59R = COOP
  GOTO 7
6 K59R = 1 - K59R
7 RETURN
  END

```

```

FUNCTION K73R(J,M,K,L,R, JA)
C BY GEORGE ZIMMERMAN
C TYPED BY JM 3/20/79
  k73r=ja      ! Added 7/27/93 to report own old value
  IF (M .GT. 1) GOTO 10
  IAGGD = 4
  IDUNU = 0
  IDUNB = 0
  IPAYB = 8
  ITEST = 1
  IPOST = 0
10 K73R = IPOST
  IF (J .NE. ITEST) RETURN
  IF (ITEST .EQ. 1) IDUNU = IDUNU + 1
  IF (ITEST .EQ. 0) IDUNB = IDUNB + 1
  IF ((IDUNU .LT. IAGGD) .AND. (IDUNB .LT. IPAYB)) RETURN
  IDUNU = 0
  IDUNB = 0
  IPOST = 0
  IF (J .EQ. 1) IPOST = 1
  K73R = IPOST
  ITEST = 0
  IF (IPOST .EQ. 0) ITEST = 1
  IF (ITEST .EQ. 0) GOTO 20
  IAGGD = IAGGD - 3 + (K / M)
  IF (IAGGD .LE. 0) IAGGD = 1
  RETURN
20 IPAYB = INT(1.6667 * FLOAT(IAGGD + 1))

```



```

        RETURN
    END
    FUNCTION K55R(J,M,K,L,R, JA)
C BY STEVE NEWMAN
C TYPED BY J|M
    k55r=ja      ! Added 7/27/93 to report own old value
    IF (M .NE. 1) GOTO 10
C INITIAL BELEIFIS
    ALPHA = 1.0
    BETA = 0.0
    IOLD = 0
    QCA = 0
    QNA = 0
    QCB = 0
    QNB = 0
    MUTDEF = 0
C UPDATE STATS OF HIS CONTINGENCIES
10    IF (M .LE. 2) GOTO 30
        IF (IOLD .EQ. 1) GOTO 20
        IF (J .EQ. 0) QCA = QCA + 1
        QNA = QNA + 1
        ALPHA = QCA / QNA
        GOTO 30
20    IF (J .EQ. 0) QCB = QCB + 1
        QNB = QNB + 1
        BETA = QCB / QNB
C SAVE OWN PAST
30    IOLD = K55R
C CALCULATE RELATIVE EXPECTATIONS OF POLICIES
C DEFECT GIVES 0
        POLC = 6 * ALPHA - 9 * BETA - 2
        POLALT = 4 * ALPHA - 6 * BETA - 1
        IF (POLC .GE. 0) GOTO 40
        IF (POLALT .GE. 0) GOTO 70
        GOTO 60
40    IF (POLC .GE. POLALT) GOTO 50
        GOTO 70
C POLC BEST, COOPERATIVE
50    K55R = 0
        RETURN
C BEST TO DEFECT
60    K55R = 1
        IF (J .EQ. 0 .OR. IOLD .EQ. 0) GOTO 100
        MUTDEF = MUTDEF + 1
        IF (MUTDEF .GT. 3) GOTO 110
        RETURN
110   K55R = 0
        RETURN
100   MUTDEF = 0
        RETURN
C POLALT BEST, ALTERNATE C AND D
70    K55R = 1 - K55R
        RETURN
    END
    FUNCTION K81R(J,M,K,L,R, JA)
C BY MARTYN JONES
C EDITED FROM BASIC BY AX 3/25/79

```

```

C TYPED BY JM 3/27/79, COR BY AX 3/28/79
  INTEGER C,T4,T5
  REAL L4(8,2)
  DIMENSION X(8)
k81r=ja      ! Added 7/27/93 to report own old value
  IF (M .EQ. 81 .AND. K .EQ. L .AND. K .EQ. 237) T0 = 1
  IF (M .NE. 1) GOTO 555
  DO 535 C = 1,8
    L4(C,1) = 0
535  L4(C,2) = 0
    T0 = 0
    T4 = 0
    T5 = 0
    T6 = 25
    T8 = 0
    T9 = 5
    D4 = 0
    A = 0
    B = 0
    S1 = 0
    DO 9997 C = 1,8
9997  X(C)=0
555  IF (M .EQ. 2 .AND. J .EQ. 1) T9 = 9
    IF (M .LT. T9) GOTO 800
    IF (T5 .GT. 7) T5 = T5 - 8
    IF (J .EQ. 0) L4(T5+1,1) = L4(T5+1,1) + 1
    IF ((T9 .EQ. 9) .AND. (T0 .EQ. 1)) GOTO 1270
    GOTO 1020
564  IF (L .GT. K + T6) GOTO 800
    D4 = T4
    IF (D4 .GT. 7) D4 = D4 - 8
c put gosub 1200 here to avoid compiler error 7/29/93
    A1 = L4(D4+1,1)
    A2 = L4(D4+1,2)
    IF (A2 .EQ. 0) A2 = 1
    A3 = A1 / A2
    A = 3 * A3
    B = A + A3 + 1
610 DO 630 C = 1,4
    X(C) = A
630  X(C + 4) = B
    E0 = 5
    E1 = 6
    E2 = 7
    E3 = 8
    F0 = 3
    F1 = 4
    F2 = 7
    F3 = 8
    L900 = 1
    GOTO 900
670  E0 = 3
    E1 = 4
    F0 = 2
    F2 = 6
    L900 = 2
    GOTO 900

```

```

710 GOTO 1100
720 K81R = 1
    IF (S1 .LT. 5) K81R = 0
    GOTO 810
800 K81R = J
810 T5 = T4
    IF ((M/10) * 10 .EQ. M) GOTO 860
815 IF (T4 .GT. 7) T4 = T4 - 8
    IF (M .GT. 3) L4(T4+1,2) = L4(T4+1,2) + 1
    IF (T4 .GT. 4) T4 = T4 - 4
    T4 = T4 * 2 + K81R
    RETURN
860 DO 880 C = 1,8
    L4(C,1) = L4(C,1) * 9
880 CONTINUE
    T6 = T6 + 1
    GOTO 815
900 IF (T4 .GT. 4) T4 = T4 - 4
    T4 = T4 * 2
    DO 1000 C = 1,8
    D4 = T4
    IF (C .EQ. E0 .OR. C .EQ. E1 .OR. C .EQ. E2 .OR. C .EQ. E3)
+D4 = T4 + 1
    IF (D4 .EQ. 9) D4 = 1
    IF (D4 .GT. 7) D4 = D4 - 8
c put gosub 1200 here Ax 7/29/93
    A1 = L4(D4+1,1)
    A2 = L4(D4+1,2)
    IF (A2 .EQ. 0) A2 = 1
    A3 = A1 / A2
    A = 3 * A3
    B = A + A3 + 1
960 IF (C .EQ. F0 .OR. C .EQ. F1 .OR. C .EQ. F2 .OR. C .EQ. F3)
+GOTO 990
    X(C) = X(C) + A
    GOTO 1000
990 X(C) = X(C) + B
1000 CONTINUE
    GOTO (670,710), L900
1020 IF (J .NE. 1) GOTO 1025
    T8 = T8 + 1
    GOTO 1070
1025 IF (.NOT.(T8 .GE. 0. .AND. T8 .LT. 6.)) GOTO 1030
    T8 = 0
    GOTO 564
1030 IF (T8 .GT. 0) T8 = -200
    K81R = 0
    T8 = T8 + 1
    GOTO 810
1070 IF (T8 .LT. 8 .OR. T8 .GT. 9) GOTO 1080
    K81R = 0
    GOTO 810
1080 IF (T8 .GT. 1) T8 = 1
    GOTO 564
1100 S = 0
    DO 1150 C = 1,8
    IF (X(C) .LE. S) GOTO 1150

```

```

      S = X(C)
      S1 = C
1150  CONTINUE
      GOTO 720
c moved "GOSUB1200" in proper places to avoid compiler error.7/29/93
1270  IF (J .NE. 1) GOTO 1272
      T0 = 0
      GOTO 1020
1272  T2 = 0
1275  IF (.NOT.((M .GT. 80 + T2) .AND. (M .LT. 140 + T2))) GOTO 1280
      K81R = 1
      GOTO 810
1280  IF (.NOT.((M .GE. 140 + T2) .AND. (M .LE. 180 + T2))) GOTO 1285
      K81R = 0
      GOTO 810
1285  T2 = T2 + 100
      GOTO 1275
      END
      FUNCTION K87R(J,M,K,L,R, JA)
C BY E E H SCHURMANN
C EDITED FROM BASIC BY AX 3/25/79
C TYPED BY JM 3/31/79
      k87r=ja      ! Added 7/27/93 to report own old value
      IF (M .EQ. 1) GOTO 695
      S = 2 * J + H + 1
      IF (Z .EQ. 1) GOTO 630
      IF (J .EQ. 0) GOTO 692
      Z = 1
630   IF (S .GT. 1) GOTO 650
      Q6 = Q6 * .57 + .43
      GOTO 680
650   IF (S .EQ. 4) GOTO 670
      Q6 = .5 * Q6
      GOTO 680
670   Q6 = .74 * Q6 + .104
680   K87R = 1
      H = 1
      IF (R .GT. Q6) RETURN
692   K87R = 0
      H = 0
      RETURN
695   Z = 0
      Q6 = .5
      S = 0
      K87R = 0
      H = 0
      RETURN
      END
      FUNCTION K53R(J,M,K,L,R, JA)
C BY HENRY NUSSBACHER 1/30/79
C TYPED BY JM
      INTEGER C(10),D,Z
      k53r=ja      ! Added 7/27/93 to report own old value
510   IF (M .GT. 10) GOTO 610
512   C(M) = J
520   GOTO 810
C NOW CHECK ON PLAYER'S PREVIOUS 10 MOVES

```

```

610 D = 0
611 DO 613 Z = 2,10
612     C(Z-1) = C(Z)
613 CONTINUE
614 C(10) = J
620 DO 650 Z = 1,10
630     IF (C(Z) .EQ. 0) GOTO 650
640     D = D + 1
650 CONTINUE
700 IF (D .GT. 8.9) GOTO 730
705 IF (D .EQ. 8) GOTO 745
710 IF (D .EQ. 7) GOTO 780
715 IF (D .EQ. 6) GOTO 780
720 IF (D .EQ. 5) GOTO 780
732 IF (D .EQ. 4) GOTO 745
725 IF (D .EQ. 3) GOTO 745
726 IF (D .EQ. 2) GOTO 780
727 IF (D .EQ. 1) GOTO 782
728 IF (D .EQ. 0) GOTO 810
730 IF (R .LT. .94) GOTO 830
740 GOTO 810
745 IF (R .LT. .915) GOTO 830
755 GOTO 810
780 IF (R .LT. .87) GOTO 830
781 GOTO 810
782 IF (R .LT. .23) GOTO 830
810 K53R = 0
811 RETURN
830 K53R = 1
831 RETURN
      END
      FUNCTION K65R(J,M,K,L,R, JA)
C BY MARK F. BATELL
C TYPED BY JM 3/15/79
      k65r=ja      ! Added 7/27/93 to report own old value
      IF (M .EQ. 1) GOTO 10
      GOTO 20
10     LASTD = 0
      DIFF = 0
      TOTD = 0
      K65R = 0
      GOTO 100
20     IF (TOTD .GE. 10) GOTO 90
      IF (J .EQ. 1) GOTO 30
      K65R = 0
      GOTO 100
30     TOTD = TOTD + 1
      IF (TOTD .GE. 10) GOTO 90
      IF (LASTD .EQ. 0) GOTO 40
      DIFF = M - LASTD
      IF (DIFF .LE. 4) GOTO 80
40     LASTD = M
      K65R = 0
      GOTO 100
80     TOTD = 10
90     K65R = 1
100    RETURN

```

```

END
FUNCTION K34R(J,M,K,L,R, JA)
C BY JAMES W. FRIEDMAN
C TYPED FROM FORTRAN BY AX, 1.17,79
  k43r=ja      ! Added 7/27/93 to report own old value
  K34R=0
  IF(M.EQ.1) JT=0
  JT=JT+J
  IF(JT.GT.0) K34R=1
  RETURN
END

C=====
C Not nice rules in second round of tour (cut and pasted 7/15/93)
FUNCTION K75R(J,M,K,L,R,JA)
C BY P D HARRINGTON
C TYPED BY JM 3/20/79
  DIMENSION HIST(4,2),ROW(4),COL(2),ID(2)
  K75R=JA      ! Added 7/32/93 to report own old value
  IF (M .EQ. 2) GOTO 25
  IF (M .GT. 1) GOTO 10
  DO 5 IA = 1,4
  DO 5 IB = 1,2
5  HIST(IA,IB) = 0
  IBURN = 0
  ID(1) = 0
  ID(2) = 0
  IDEF = 0
  ITWIN = 0
  ISTRNG = 0
  ICOOP = 0
  ITRY = 0
  IRDCHK = 0
  IRAND = 0
  IPARTY = 1
  IND = 0
  MY = 0
  INDEF = 5
  IOPP = 0
  PROB = .2
  K75R = 0
  RETURN
10 IF (IRAND .EQ. 1) GOTO 70
  IOPP = IOPP + J
  HIST(IND,J+1) = HIST(IND,J+1) + 1
  IF (M .EQ. 15 .OR. MOD(M,15) .NE. 0 .OR. IRAND .EQ. 2) GOTO 25
  IF (HIST(1,1) / (M - 2) .GE. .8) GOTO 25
  IF (IOPP * 4 .LT. M - 2 .OR. IOPP * 4 .GT. 3 * M - 6) GOTO 25
  DO 12 IA = 1,4
12 ROW(IA) = HIST(IA,1) + HIST(IA,2)
  DO 14 IB = 1,2
  SUM = .0
  DO 13 IA = 1,4
13 SUM = SUM + HIST(IA,IB)
14 COL(IB) = SUM
  SUM = .0
  DO 16 IA = 1,4
  DO 16 IB = 1,2

```

```

EX = ROW(IA) * COL(IB) / (M - 2)
IF (EX .LE. 1.) GOTO 16
SUM = SUM + ((HIST(IA,IB) - EX) ** 2) / EX
16 CONTINUE
IF (SUM .GT. 3) GOTO 25
IRAND = 1
K75R = 1
RETURN
25 IF (ITRY .EQ. 1 .AND. J .EQ. 1) IBURN = 1
IF (M .LE. 37 .AND. J .EQ. 0) ITWIN = ITWIN + 1
IF (M .EQ. 38 .AND. J .EQ. 1) ITWIN = ITWIN + 1
IF (M .GE. 39 .AND. ITWIN .EQ. 37 .AND. J .EQ. 1) ITWIN = 0
IF (ITWIN .EQ. 37) GOTO 80
IDEF = IDEF * J + J
IF (IDEF .GE. 20) GOTO 90
IPARTY = 3 - IPARTY
ID(IPARTY) = ID(IPARTY) * J + J
IF (ID(IPARTY) .GE. INDEF) GOTO 78
IF (ICOOP .GE. 1) GOTO 80
IF (M .LT. 37 .OR. IBURN .EQ. 1) GOTO 34
IF (M .EQ. 37) GOTO 32
IF (R .GT. PROB) GOTO 34
32 ITRY = 2
ICOOP = 2
PROB = PROB + .05
GOTO 92
34 IF (J .EQ. 0) GOTO 80
GOTO 90
70 IRDCHK = IRDCHK + J * 4 - 3
IF (IRDCHK .GE. 11) GOTO 75
K75R = 1
RETURN
75 IRAND = 2
ICOOP = 2
K75R = 0
RETURN
78 ID(IPARTY) = 0
ISTRNG = ISTRNG + 1
IF (ISTRNG .EQ. 8) INDEF = 3
80 K75R = 0
ITRY = ITRY - 1
ICOOP = ICOOP - 1
GOTO 95
90 ID(IPARTY) = ID(IPARTY) + 1
92 K75R = 1
95 IND = 2 * MY + J + 1
MY = K75R
RETURN
END
FUNCTION K47R(J,M,K,L,R,JA)
C BY RICHARD HUFFORD
C TYPED BY JM
INTEGER NUM,DEN,RF,DEF,COOP,LONG,SHORT,SH2(5)
K47R=JA ! Added 7/32/93 to report own old value
IF (M .GT. 1) GOTO 100
C INITIALIZE
NUM = 2

```

```

DEN = 2
RF = 20
DEF = 1
COOP = 0
LONG = 1
SHORT = 5
DO 10 N = 1,5
SH2(N) = 1
10 CONTINUE
N = 1
MYLAST = 0
MYMOVE = 0
100 IF ((M .LE. RF) .AND. (J .EQ. DEF)) RF = M + (20 * NUM) / DEN + 1
C DETERMINE OPPONENT'S LONG AND SHORT TERM SENSE
200 N = MOD(N,4) + 1
SHORT = SHORT - SH2(N)
IF (J .EQ. MYLAST) GOTO 500
SH2(N) = 0
GOTO 1000
500 LONG = LONG + 1
SHORT = SHORT + 1
SH2(N) = 1
1000 MYLAST= MYMOVE
C MOVE
MYMOVE = J
IF ((LONG .LT. .625 * M) .OR. (SHORT .LT. 3)) MYMOVE = DEF
IF ((LONG .GT. .9 * M) .AND. (SHORT .EQ. 5)) MYMOVE = COOP
C SHOULD I RF HOM THIS TURN
IF (M .EQ. RF) MYMOVE = DEF
IF (M .LT. RF + 2) GOTO 2000
C I RF-D HIM 2 TURNS AGO. MUST NOT GET IN A FIGHT OVER NOTHING
MYMOVE = COOP
C DETERMINE SUCCESS OF RF
NUM = NUM + J
DEN = DEN + 1 - J
C DETERMINE NEXT TURN TO RF HIM
RF = M + (20 * NUM) / DEN + 1
2000 K47R = MYMOVE
RETURN
END
FUNCTION K51R(J,M,K,L,R,JA)
C BY JOHN WILLIAM COLBERT
C TYPED BY JM
K51R=JA ! Added 7/32/93 to report own old value
IF (M .GT. 8) GOTO 5
K51R = 0
IF (M .EQ. 6) K51R = 1
LASTI = 0
GOTO 10
5 K51R = 0
LASTI = LASTI - 1
IF (LASTI .EQ. 3) K51R = 1
IF (LASTI .GT. 0) GOTO 10
IF (J .EQ. 1) K51R = 1
IF (J .EQ. 1) LASTI = 4
10 RETURN
END

```



```

        FUNCTION K78R(J,M,K,L,R,JA)
C BY FRED MAUK
C TYPED BY AX, 3/27/79 (SAME AS ROUND ONE GRAASKAMP)
    INTEGER GRASR
c Time parameter elminated Ax 7/93
    K78R=GRASR(J,M,K,L,R,JA)
    RETURN
    END
    FUNCTION K39R(J,M,K,L,R,JA)
C     BY TOM ALMY (FROM HIS PAPER TAPE)
C     EDITED BY AX, 1.16.79
    IMPLICIT INTEGER(A-Z)
    REAL R
    DIMENSION OK(3)
    K39R=JA      ! Added 7/32/93 to report own old value
cc ax test
c  write(6,77) m, step, substp
c77 format(' test k39r. m, step, substp', 3i3)
    IF(M.NE.1) GOTO 10
    STEP=1
    SUBSTP=1
    BOTHD=0
    TITCNT=0
    TATCNT=0
    EVIL=0
    N=1
    F=0
    DO 1 I=1,3
    OK(I)=0
1     CONTINUE
    TOTK=0
    OLDMOV=0
10    CONTINUE
C     DO TABULATION
    IF(K39R+J.EQ.2) BOTHD=BOTHD+1
    IF(K39R+J.LT.2) BOTHD=0
    COUNT=COUNT-1
    K39R=0
    VOLDMV=OLDMOV
    OLDMOV=J
    IF(J.EQ.1) TATCNT=TATCNT+1
    IF(EVIL.EQ.0 .AND. J.EQ.1) EVIL=1
20    CONTINUE
    GOTO (100,200,300,400,500), STEP
C     PLAY TIT FOR TWO TATS
100   CONTINUE
    GOTO(101,110,120), SUBSTP
C INITIALIZE ALL DEFENSIVE MODES
C     OK AND TOTK NOT RESET IN ORDER TO BIAS TOWARDS KEEPING
C     THIS PLAY MODE IF WE HAVE JUST FINISHED EXPLOITING.
101   CONTINUE
    COUNT=10
    TATCNT=0
    TITCNT=0
    SUBSTP=2
    GOTO 20
C PLAY TIT FOR TWO TATS

```

```

110     CONTINUE
      IF((VOLDMV+OLDMOV).EQ.2) K39R=1
      TITCNT=TITCNT+K39R
      IF(COUNT.EQ.0) SUBSTP=3
      RETURN
C EVALUATE PLAY
120     CONTINUE

cc ax test
c  if (m.eq. 51) write(6,7120) m, step, substp
c7120  format(' test 7120 after 120. m, step, substp', 3i3)
      OLDSTP=STEP
      OK(STEP)=K-TOTK
      TOTK=K
      SUBSTP=1
      IF(TATCNT.GT.0) GOTO 130
C     NICE OPPONENT--TRY TO TAKE ADVANTAGE!
      STEP=4
C     IF OPPONENT NOT REALLY NICE--DON'T TRY TO TAKE DVANTAGE
      IF (EVIL.EQ.1) STEP=1
      IF (EVIL.EQ.0) EVIL=-1
      GOTO 20
130     CONTINUE
C     LET US FIND BEST DEFENSE (HIGHEST SCORE)
      STEP=1
      DO 150 I1=1,2
      DO 150 I2=2,3

cc ax test
c  if (m.eq. 51) write(6,71302) m, step, substp, i1, i2, ok(i1), ok(i2)
c71302 format(' test 71302 After 130. m, step, substp, i1, i2, ok(i1), ok(I2)', 7i3)
      IF(OK(I1).EQ.0.OR. OK(I2).EQ.0) GOTO 150
      IF(OK(I1).GE.OK(I2)) GOTO 150
      IF(STEP.EQ.I1) STEP=I2
150     CONTINUE
C     ADVANCE TO NEW STEP IF NEXT ONE NOT TESTED AND EITHER PONENT
C     IS VERY NASTY OR IS EXPLOITING US
c next 2 lines are test4 added by Ax 7/23
c  if (step .gt. 2) write(6, 737) j, m, k, l, step, substp
c737  format(' test737 from K39r. j,m,k,l,step, substp: ', 6i4)
c Next statement broken up to prevent complier error. Two clauses separated.Ax 7/26/93
c  IF (STEP.NE.3 .AND. OK(STEP+1).EQ.0 .AND.
c    1(TATCNT.GE.4 .OR. TITCNT.EQ.0))
c    1 STEP=STEP+1
      if (step.eq.3) goto 777      ! if step=3 skip next test
      IF ( (OK(STEP+1).EQ.0) .AND.
      1(TATCNT.GE.4 .OR. TITCNT.EQ.0))
      1 STEP=STEP+1
777  continue
C     IF WE PUNISHED TOO SEVERLY, THEN GO ALL C TO ECOOPERATE
cc ax test
c  if (m.eq. 51) write(6,747) m, step, substp
c747  format(' test 747 k39r After 737. m, step, substp', 3i3)
      IF(STEP.LT.OLDSTP .AND. BOTHD .GT.0) STEP=5
      GOTO 20
C     PLAY TIT FOR TAT
200     CONTINUE
      GOTO (101,210,120), SUBSTP

```

```

210     CONTINUE
      IF(OLDMOV.EQ.1) K39R=1
      TITCNT=TITCNT+K39R
      IF (COUNT.EQ.0) SUBSTP=3
      RETURN
C     PLAY ALL DEFECTS
300     CONTINUE
      GOTO (101,310,120), SUBSTP
310     CONTINUE
cc ax test
c  if (m.eq. 51) write(6,7727) m, step, substp
c7727  format(' test 7727. m, step, substp', 3i3)
      K39R=1
      TITCNT=TITCNT+1
      IF (COUNT.EQ.0) SUBSTP=3
      RETURN
C     EXPLOIT
400     CONTINUE
      GOTO(401,402,403,404), SUBSTP
C     DO A DISRUPT
401     CONTINUE
      SUBSTP=2
      K39R=1
      COUNT=N
      TATCNT=0
      RETURN
C     COOPERATE FOR A WHILE
402     CONTINUE
      IF(COUNT.EQ.0) SUBSTP=3
      RETURN
C     DECIDE WHAT TO DO
403     CONTINUE
      IF(TATCNT.NE.0) GOTO 410
C     WE HAVEN'T BEEN PUNISHED--TRY IT AGAIN
      F=1
      GOTO 401
C     WE HAVE BEEN PUNISHED--DECIDE ACTION
410     CONTINUE
      IF(F.EQ.0) GOTO 420
C     WE HAD BEEN RUNNING  -TRY LATER WITH LARGER GAP
      N=N+1
      SUBSTP=1
      STEP=1
      GOTO 20
C     TOUCHY PROGRAM--COOPERATE UNTIL DEFECTION THEN RESUME FOR 2T
420     CONTINUE
      SUBSTP=4
      IF(J.EQ.1) N=N+1
      TATCNT=J
      RETURN
C     COOP UNTIL DEFECTION
404     CONTINUE
C     ALLOW A GROTESQUE PUNISHMENT (5 TATS WITHOUT US EFECTING)
      IF(TATCNT.LE.4) RETURN
      SUBSTP=1
      STEP=1
      GOTO 20

```

```

C      DO ALL C FOR 5 MOVES TO COOL THINGS OFF
500    CONTINUE
      IF(SUBSTP.EQ.2) GOTO 520
      COUNT=5
      SUBSTP=2
520    CONTINUE

cc ax test
c      if (m.eq. 51) write(6,7520) m, step, substp
c7520  format(' test 7520 after 520. m, step, substp', 3i3)
      IF(COUNT.NE.0) RETURN
      SUBSTP=1
      GOTO 130
      END
      FUNCTION K67R(J,M,K,L,R,JA)
C EDITED FROM BASIC FROM AX. 3/10/79
C TYPED BY JM 3/16/80
C BY CRAIG FEATHERS
      REAL NO,NK
      K67R=JA      ! Added 7/32/93 to report own old value
      IF (M .NE. 1) GOTO 510
      S = 0
      AD = 5
      NO = 0
      NK = 1
      AK = 1
      FD = 0
      C = 0
510    IF (FD .NE. 2) GOTO 520
      FD = 0
      NO = (NO * NK + 3 - 3 * J + 2 * K67R - K67R * J) / (NK + 1)
      NK = NK + 1
520    IF (FD .NE. 1) GOTO 530
      FD = 2
      AD = (AD * AK + 3 - 3 * J + 2 * K67R - K67R * J) / (AK + 1)
      AK = AK + 1
530    IF (J .EQ. 0) GOTO 540
      S = S + 1
      GOTO 545
540    S = 0
      C = C + 1
545    K67R = 0
      IF (ABS(FD - 1.5) .EQ. .5) GOTO 599
      IF (K .LT. 2.25 * M) GOTO 575
      P = .95 - (AD + NO - 5) / 15 + 1. / M**2 - J / 4.
      IF (R .LE. P) GOTO 599
      K67R = 1
      FD = 1
      GOTO 599
575    IF (K .LT. 1.75 * M) GOTO 595
      P = .25 + C / M - S * .25 + (K - L) / 100. + 4. / M
      IF (R .LE. P) GOTO 599
      K67R = 1
      GOTO 599
595    K67R = J
599    RETURN
      END

```

```

FUNCTION K69R(J,M,K,L,R,JA)
C BY JOHANN JOSS
C EDITED FROM BASIC BY AX, 3/10/79
C TYPED BY JM 3/16/79
      K69R=JA      ! Added 7/32/93 to report own old value
      IF (M .EQ. 1) GOTO 600
      IF (J .EQ. 1) GOTO 512
      C = C + 1
512   IF (S .EQ. 1) GOTO 700
      IF (S .EQ. 2) GOTO 800
      IF (S .EQ. 3) GOTO 900
      IF (S .EQ. 4) GOTO1000
      IF (S .EQ. 5) GOTO 1100
600   S = 1
      F = 0
      D = 0
      C = 0
      K69R = 0
      RETURN
700   IF (R .LT. 0.1) GOTO 720
702   IF (J .EQ. 0) GOTO 708
      D = D + 1
      GOTO 710
708   D = 0
710   IF (D .GT. 20) GOTO 820
      IF (C .LT. 0.7 * (M - 3)) GOTO 800
      K69R = J
      RETURN
720   S = 5
      K69R = 1
      RETURN
800   S = 2
      IF (J .EQ. 0) GOTO 808
      D = D + 1
      GOTO 810
808   D = 0
810   IF (D .GT. 10) GOTO 830
      K69R = 1
      RETURN
820   S = 3
      K69R = 0
      D = 0
      RETURN
830   S = 3
      K69R = 1
      RETURN
900   IF (J .EQ. 0) GOTO 908
      D = D + 1
      GOTO 910
908   D = 0
910   IF (D .GT. 20) GOTO 820
      K69R = J
      RETURN
1000  IF (J .EQ. 0) GOTO 1006
      F = F + 1
      IF (F .GT. 3) GOTO 820
1006  S = 1

```

```

        K69R = 0
        RETURN
1100  S = 4
        GOTO 702
        END
        FUNCTION K43R(J,M,K,L,R,JA)
C BY R.D. ANDERSON
C TYPED FROM FORTRAN BY AX, 1/25/79
        K43R=JA      ! Added 7/32/93 to report own old value
        IF(M.GT.1)GOTO 1
        NCC=0
        NCD=0
        NDC=0
        NDD=0
        KOUNT=0
        MYTWIN=0
        GOTO 900
1      IF(M.LT.3)GOTO 3
        IF(IOLD2.EQ.1)GOTO 2
        NCC=NCC+1-J
        NCD=NCD+J
        GOTO 3
2      NDC=NDC+1-J
        NDD=NDD+J
3      IOLD2=IOLD1
        IF(M.GE.16)GOTO 4
        IF(J.EQ.0)GOTO 900
        IF(KOUNT.GE.3) GOTO 900
        KOUNT=KOUNT+1
        GOTO 901
4      IF(M.EQ.17.AND.J.EQ.1.AND.NCD.EQ.1.AND.NDD.EQ.0) MYTWIN=1
        IF((NCD*3).GE.(NCC+NCD))GOTO 901
        IF(M.NE.(4*(M/4))) GOTO 900
        IF(MYTWIN.EQ.1) GOTO 900
        IF (NDC.GE.(M/12).OR.NDD.EQ.0) GOTO 901
900    IOLD1=0
        GOTO 999
901    IOLD1=1
999    K43R = IOLD1
        RETURN
        END
        FUNCTION K76R(J,M,K,L,R,JA)
C BY DAVID GLADSTEIN
C FROM CARDS BY JM 3/16/79
        LOGICAL PATSY
        K76R=JA      ! Added 7/32/93 to report own old value
        IF (M .NE. 1) GO TO 1
        PATSY = .TRUE.
        DC = 0
        MDC = 0
        G = 1
        K76R = 1
        RETURN
1      IF (PATSY) GO TO 2
        K76R = J
        RETURN
2      IF (J .NE. 1) GO TO 3

```

```

PATSY = .FALSE.
K76R = 0
RETURN
3   DC = DC + 1
    IF (G .EQ. 0) MDC = MDC + 1
    G = 0
    IF (MDC / (DC + 1) .GE. .5) G = 1
    K76R = G
    RETURN
    END
    FUNCTION K52R(J,M,K,L,R,JA)
C BY DAVID A. SMITH
C EDITED FROM BASIC BY AX,2/11/79
C TYPED BY JM
    INTEGER D8,D9
    K52R=JA      ! Added 7/32/93 to report own old value
    K52R = 0
    IF (M .GT. 1) GOTO 305
    D9 = 0
    D8 = 0
305 D9 = D9 + 1
    IF (J .GT. 0) GOTO 320
    D9 = 0
320 IF (D9 .LT. 2) GOTO 345
    K52R = 1
    IF (D9 .LT. (5+ 3*D8)) GOTO 345
    D9 = 0
    D8 = D8 + 1
345 IF (R .GT. .05) GOTO 355
    K52R = 1 - K52R
355 RETURN
    END
    FUNCTION K82R(J,M,K,L,R,JA)
C BY ROBERT A LEYLAND
C TYPED BY JM 3/22/79
    K82R=JA      ! Added 7/32/93 to report own old value
    K82R = J
    IF (M .EQ. 1) GOTO 2180
    I5 = I5 + J
    D4 = D4 + J
    IF (J .EQ. 0 .AND. I5 .GT. 1) GOTO 2100
2010 IF (M .LT. 30) RETURN
    IF (I3 .EQ. 0) GOTO 2190
    IF (ABS(D4/(M - 1.0) - 0.5) .LT. 0.1) X = X - 0.2
    IF (I2 .EQ. 1) GOTO 2070
2030 IF (R .GT. X) GOTO 2150
    I2 = I1
    RETURN
2070 IF (J .EQ. 0) GOTO 2120
    X = X + 0.15
    IF (X .GT. 1.0) X = 1.0
    GOTO 2190
2100 IF (I5 .GT. 5) GOTO 2200
    I5 = 0
    GOTO 2010
2120 X = X - 0.05
    IF (X .LT. 0.0) X = 0.0

```

```

        I2 = 0
        IF (X .GE. 0.3) RETURN
        GOTO 2030
2150   K82R = 1
        I1 = 1
        RETURN
2180   X = 0.75
        I5 = 0
        D4 = 0.0
2190   I2 = 0
        I3 = 1
        IF (I5 .GT. 5) I3 = 0
2200   I5 = 0
        I1 = 0
        K82R = 0
        RETURN
        END
        FUNCTION K45R(J,M,K,L,R,JA)
C BY MICHAEL F. MCGURRIN
C TYPED FROM FORTRAN BY AX, 1/26/79
        K45R=JA      ! Added 7/32/93 to report own old value
        IF(M.GT.3) GOTO 40
        IF(M.NE.1) GOTO 10
        JOLD=0
        A=0
        B=0
        C=0
        E=0
        K45R=1
        RETURN
10     IF(M.NE.2) GOTO 20
        IF(J.EQ.1) GOTO 30
        K45R=0
        D=0
        RETURN
30     K45R=0
        D=1
        RETURN
20     IF(J.EQ.1) GOTO 50
        IF(D.EQ.1) GOTO 60
        K45R=0
        A=1
        RETURN
60     K45R=0
        RETURN
50     K45R=0
        IF(D.EQ.1) C=1
        RETURN
40     IF(C.EQ.1) GOTO 70
        IF(B.EQ.1) GOTO 80
        IF(A.EQ.1) GOTO 90
        IF(D.EQ.1) GOTO 120
        IF(J.EQ.1) GOTO 100
        K45R=0
        B=1
        RETURN
100    K45R=0

```



```

      C=1
      RETURN
120   IF(J.EQ.1) GOTO 130
      K45R=0
      B=1
      RETURN
130   K45R=1
      C=1
      RETURN
70    K45R=J
      RETURN
80    K45R=0
      IF((JOLD.EQ.1).AND.(J.EQ.1)) K45R=1
      JOLD=J
      RETURN
90    K45R=1
      E=E+1
      IF(E.NE.8) GOTO 110
      E=0
      JOLD=J
      RETURN
110   IF(.NOT.((JOLD.EQ.1).AND.(J.EQ.1))) K45R=0
      JOLD=J
      RETURN
      END

      FUNCTION K62R(J,M,K,L,R,JA)
C BY HOWARD R HOLLANDER
C TYPED BY JM 2/25/79
      K62R=JA          ! Added 7/32/93 to report own old value
      IF (M .NE. 1) GOTO 505
      JOLD = 0
      IRAN = 23 * R + 1
505   K62R = 0
      IF (M .NE. IRAN) GOTO 510
      K62R = 1
      IRAN = 23 * R + M + 1
      GOTO 515
510   IF ((JOLD .EQ. 1) .AND. (J .EQ. 1)) K62R = 1
515   JOLD = J
      RETURN
      END

      FUNCTION K48R(J,M,K,L,R,JA)
C BY GEORGE HUFFORD
C TYPED BY JM
      DIMENSION IARRAY(5),IPO2(5)
C NOT NICE, DETERMINISTIC, FORGIVING
      DATA IPO2/2,4,3,5,1/
      K48R=JA          ! Added 7/32/93 to report own old value
      IF (M .EQ. 1) GOTO 1
      IF (M .LE. 5) GOTO 2
      MM = MOD(M-1,5) + 1
      K48R = IARRAY(MM)
      IF (MM .NE. 1) RETURN
      KOLD = K5
      K5 = K - KLAST
      KLAST = K
      IF (KOLD .GT. K5) ICHAN = -ICHAN

```

```

IF (KOLD .GT. K5) IPO1 = IPO1 + ICHAN
IF (IPO1 .LT. 1) IPO1 = 0
IF (IPO1 .GT. 5) IPO1 = 6
IF (IPO1 .LT. 1 .OR. IPO1 .GT. 5) RETURN
IARRAY (IPO2(IPO1)) = IARRAY(IPO2(IPO1)) + ICHAN
IPO1 = IPO1 + ICHAN
K48R = IARRAY(MM)
RETURN
1 KOLD = 0
  K5 = 0
  KLAST = 0
  DO 3 I =1,5
3 IARRAY(I) = 0
  MM = 0
  ICHAN = 1
  IPO1 = 1
2 IARRAY(IPO2(IPO1)) = J
  IPO1 = IPO1 + J
  K48R = J
  RETURN
  END
FUNCTION K50R(J,MOVN,KM,KH,R,JA)
C BY RIK
C TYPED BY JM, CORRECTED BY AX, 2/27/79
  K50R=JA      ! Added 7/32/93 to report own old value
  K50R = 0
  IF ((J .EQ. 0) .AND. (R .GE. 0.9)) K50R = 1
  RETURN
  END
FUNCTION K77R(JPICK,MOVEN,ISCORE,JSCORE,RANDOM,JA)
  DIMENSION KEXP(5)
C BY SCOTT FELD
C TYPED BY JM 3/22/79
  K77R=JA      ! Added 7/32/93 to report own old value
  IF (MOVEN .GT. 1) GOTO 6
  JSTR = 3
  KTRY = 0
  KEXP(1) = 100
  KEXP(2) = 100
  KEXP(3) = 100
  KEXP(4) = 100
  KEXP(5) = 100
  KI = 0
6 IF (KTRY .LT. 20) GOTO 9
  KEXP(JSTR) = ISCORE - KI
  IF (JSTR .EQ. 5) GOTO 7
  IF (KEXP(JSTR + 1) .LE. KEXP(JSTR)) GOTO 7
  JSTR = JSTR + 1
  GOTO 8
7 IF (JSTR .EQ. 1) GOTO 8
  IF (KEXP(JSTR - 1) .LE. KEXP(JSTR)) GOTO 8
  JSTR = JSTR - 1
  JPICK = 0
8 KI = ISCORE
  KTRY = 0
9 KTRY = KTRY + 1
  GOTO (10,20,30,40,50), JSTR

```

```

10    K77R = 0
      RETURN
20    K77R = 0
      IF (JPICK .EQ. 0) RETURN
      IF (RANDOM .LE. .75) K77R = 1
      RETURN
30    K77R = JPICK
      RETURN
40    K77R = 1
      IF (JPICK .EQ. 1) RETURN
      IF (RANDOM .LE. .75) K77R = 0
      RETURN
50    K77R = 1
      RETURN
      END
      FUNCTION K89R(HCM,MN,MYSC,HSC,RANDOM,JA)
C BY GENE SNODGRASS
C FROM CARDS BY JM 3/22/79
      IMPLICIT INTEGER(A-Q,S-Z)
      DIMENSION SC(6),SL(6),ST(5),GT(5),TM(6)
      K89R=JA      ! Added 7/32/93 to report own old value
      IF(.NOT.(MN.EQ.1))GOTO 23010
      DO 23012I=1,5
      GT(I)=0
      TM(I)=0
      SL(I)=1
23012 CONTINUE
23013 CONTINUE
      CN=10
      TM(6)=0
      SL(6)=1
      CSRC=5
      MYLM=1
      HLM=0
23010 CONTINUE
23014 CONTINUE
      CODE=CN/10
      IF(.NOT.(10*CODE.EQ.CN))GOTO 23017
      SC(CODE)=MYSC
23017 CONTINUE
      IF(.NOT.(SL(CODE).EQ.1))GOTO 23019
      CN=CN+1
      TM(CODE)=TM(CODE)+1
      GOTO(10,20,30,40,50,60),CODE
10    K89R=0
      RETURN
20    K89R=1
      RETURN
30    K89R=1-MYLM
      MYLM=K89R
      RETURN
40    IF(.NOT.(HCM.EQ.1))GOTO 23021
      K89R=1
      GOTO 23022
23021 CONTINUE
      K89R=0
23022 CONTINUE

```

```

RETURN
50  IF( .NOT.(HCM.EQ.1.AND.HLM.EQ.1))GOTO 23023
    K89R=1
    GOTO 23024
23023 CONTINUE
    K89R=0
23024 CONTINUE
    HLM=HCM
    RETURN
60  SGT=0
    DO 23025I=1,5
    ST(I)=SC(I+1)-SC(I)
    SGT=SGT+ST(I)
    GT(I)=GT(I)+ST(I)
23025 CONTINUE
23026 CONTINUE
    MEAN=SGT/CSRC
    AMEAN=9*MEAN/10
    CSRC=0
    DO 23027I=1,5
    IF( .NOT.(SL(I).EQ.1))GOTO 23029
    IF( .NOT.(ST(I).LT.AMEAN))GOTO 23031
    SL(I)=0
23031 CONTINUE
    GOTO 23030
23029 CONTINUE
    IF( .NOT.(10*GT(I)/TM(I).GT.AMEAN))GOTO 23033
    SL(I)=1
23033 CONTINUE
23030 CONTINUE
    IF( .NOT.(SL(I).EQ.1))GOTO 23035
    CSRC=CSRC+1
23035 CONTINUE
23027 CONTINUE
23028 CONTINUE
    CN=10
    GOTO 23020
23019 CONTINUE
    CN=CN+10
23020 CONTINUE
23015 GOTO 23014
    END
    FUNCTION K63R(J,M,K,L,R,JA)
C BY GEORGE DUISMAN
C EDITED FROM BASIC BY AX, 3/7/79
C TYPED BY JM 3/15/79
    K63R=JA      ! Added 7/32/93 to report own old value
    IF (M .EQ. 1) ik = 1
    ik = 1 - ik
    K63R = IK
cc test 2 lines Ax 7/93. Also rewritten by Ax 7/21/93 putting ik where K63r was
cc write (6,993) k63r
cc993  format (' test from k63r, k63r= ', i3)
C COOP ON ODD MOVES ONLY
    RETURN
    END
    FUNCTION K54R(J,K,L,M,R,JA)

```

```

C BY WILLIAM H ROBERTSON
C TYPED BY JM
C AX ADD ST.999, 3/31/79
  INTEGER OPDEF,STDEF,COOPS
  LOGICAL OKDEF,MYDEF
  K54R=JA      ! Added 7/32/93 to report own old value
  K54R = 0
  IF (M .GT. 1) GOTO 5
C SET UP INITIAL CONDITIONS
  OPDEF = 0
  STDEF = 0
  DL = .20
  COOPS = 0
  OKDEF = .TRUE.
  MYDEF = .FALSE.
  NODEF = 0
  ND = 12
  RETURN
C LOWER DEFLECTION LEVEL TO 10% ON MOVE 20
C SEE IF OPONENT DEFECTS
5   IF (M .EQ. 20) DL = .10
    IF (J .EQ. 1) GOTO 10
C OPONENT DOES NOT DEFECT
  STDEF = 0
  COOPS = COOPS + 1
  IF (FLOAT(OPDEF) .GT. FLOAT(M) * DL) GOTO 20
  IF (MOD(M,ND) .EQ. 0 .AND. OKDEF) GOTO 25
  MYDEF = .FALSE.
  RETURN
C OPONENT DEFECTS IN FIRST 4 MOVES
10  COOPS = 0
    IF (M .GT. 4) GOTO 15
    K54R = 1
    RETURN
C OPPONENT DEFECTS AFTER FIRST 4 MOVES.
C START TO KEEP TRACK OF NO. OF DEFLECTIONS
15  STDEF = STDEF + 1
    OPDEF = OPDEF + 1
    IF (MYDEF) OKDEF = .FALSE.
    IF (FLOAT(OPDEF) .GT. FLOAT(M) * DL) GOTO 20
    IF (STDEF .GT. 2) GOTO 20
    MYDEF = .FALSE.
    RETURN
C OPPONENT DEFECTS EXCESSIVELY
20  IF (20 * OPDEF .LE. COOPS * M) RETURN
    K54R = 1
    MYDEF = .FALSE.
    RETURN
C PROGRAM WILL TRY A DEFECTON
25  K54R = 1
    MYDEF = .FALSE.
    NODEF = NODEF + 1
    IF (MOD(NODEF,6) .EQ. 0) ND = ND - 1
999 IF(ND .LT. 1) ND = 1
    RETURN
    END
    FUNCTION K33R(J,M,K,L,R,JA)

```

```

C BY HAROLD RABBIE
C TYPED BY AX FROM FORTRAN, 1.17.79
C
C ASSUMES THAT THE PROBABILITY OF MY OPPONENT COOPERATING
C DEPENDS ONLY ON MY LAST TWO RESPONSES.
C DETERMINISTIC, NOT NICE
  LOGICAL TWIN
  DIMENSION COOP(4),COUNT(4),P(4),COEFF(6,4),CONST(6)
  DATA CONST/ 0.,4.,6.,6.,8.,12./
  DATA COEFF/36.,16.,0.,12.,0.,0.,
2     0.,12.,18.,12.,16.,0.,
3     0.,12.,24.,9.,16.,0.,
4     0.,0.,0.,9.,12.,48./
C INITIALISE ALL STATE VARIABLES
  K33R=JA      ! Added 7/32/93 to report own old value
  IF(M.GT.1) GO TO 2
  DO 1 JJ=1,4
    COOP(JJ)=0.
    COUNT(JJ)=0.
1 CONTINUE
  LAST1=1
  LAST2=1
  TWIN=.TRUE.
C UPDATE ESTIMATE OF RELEVANT PROBABILITY
2 IF(M.LE.2) GO TO3
  COOP(INDEX)=COOP(INDEX)+FLOAT(1-J)
  COUNT(INDEX)=COUNT(INDEX)+1
  P(INDEX)=COOP(INDEX)/COUNT(INDEX)
C COMPUTE INDEX BASED ON MY LAST TWO RESPONSES
3 INDEX=2*LAST2+LAST1+1
C IDENTIFY MY TWIN
  IF(M.EQ.1) GO TO 4
  IF(J.NE.LAST1) TWIN=.FALSE.
C USE POLICY 4 FOR 22 MOVES
4 IF(M.LE.22) GO TO 24
C COOPERATE WITH MY TWIN
  IF(TWIN) GO TO 30
C COMPUTE BEST EXPECTED SCORE OVER THE 6 DIFFERENT POLICIES
  BEST=0
  DO 10 II=1,6
    SUM=CONST(II)
    DO 11 JJ=1,4
11     SUM=SUM+COEFF(II,JJ)*P(JJ)
    IF(SUM.LE.BEST) GO TO 10
    BEST=SUM
    IPOL=II
10 CONTINUE
C EXECUTE THE BEST POLICY
  GO TO (21,22,23,24,25,26), IPOL
C DISPATCH ACCORDING TO THE LAST TWO MOVES
21 GO TO (30,30,30,30),INDEX
22 GO TO (40,30,30,30),INDEX
23 GO TO (40,30,40,30),INDEX
24 GO TO (40,40,30,30),INDEX
25 GO TO (40,40,40,30),INDEX
26 GO TO (40,40,40,40),INDEX
C COOPERATE

```

```

30 K33R=0
   GO TO 50
C DEFECT
40 K33R=1
C UPDATE HISTORY
50 LAST2=LAST1
   LAST1=K33R
   RETURN
   END
   FUNCTION K71R(J,M,K,L,R,JA)
C BY JAMES E HILL
C TYPED BY JM 3/16/79
   K71R=JA      ! Added 7/32/93 to report own old value
   IF (M .EQ. 1) GOTO 1700
   IF (M .EQ. 2) GOTO 1600
   IF (J .EQ. 0) GOTO 1000
   IB = IB + 1
   IF (IB .EQ. 2) GOTO 500
500  K71R = 0
   K71R = 1
   IB = 0
   GOTO 1710
1000 IA = IA + 1
   IF (IA .EQ. 2) GOTO 110
   K71R = 0
   GOTO 1710
110  K71R = 1
   IA = 0
   GOTO 1710
1600 K71R = 1
   IF (J .EQ. 1) K71R = 0
   GOTO 1710
1700 IA = 0
   IB = 0
   K71R = 0
1710 RETURN
   END

```

cc Here's mod version of k74, early mod version follows with XX after name

```

FUNCTION K74R(J,M,K,L,R,JA)
C BY EDWARD FRIEDLAND
C TYPED BY JM 3/20/79
c temp output
   K74R=JA      ! Added to get self reported
   IF (M .NE. 1) GOTO 9
   ALPHA = 1.0
   BETA = .3
   IOLD = 0
   QCA = 0
   QNA = 0
   QCB = 0
   QNB = 0
   K74R = 0
   JSW = 0
   JS4 = 0
   JS11 = 0
   JR = 0

```

```

      JL = 0
      JT = 0
      JSM = 1
9     IF (JR .NE. 1) GOTO 10
      K74R = 1
      RETURN
10    IF (M .LE. 2) GOTO 30
      IF (IOLD .EQ. 1) GOTO 20
      IF (J .EQ. 0) QCA = QCA + 1
      QNA = QNA + 1
      ALPHA = QCA / QNA
      QCA = QCA * .8
      QNA = QNA * .8
      GOTO 30
20    IF (J .EQ. 0) QCB = QCB + 1
      QNB = QNB + 1
      BETA = QCB / QNB
      QCB = QCB * .8
      QNB = QNB * .8
30    IOLD = K74R
C CHECK FOR RANDOM
      IF (M .EQ. 37) GOTO 80
      IF (M .GT. 37) GOTO 15
      IF (M .EQ. 1) GOTO 15
      IF (J .EQ. JL) JSM = JSM + 1
      IF (JSM .GE. 3) JS4 = 1
      IF (JSM .GE. 11) JS11 = 1
      IF (J .NE. JL) JSW = JSW + 1
      JSM = 1
      JT = JT + J
15    POLC = 6 * ALPHA - 8 * BETA - 2
      POLALT = 4 * ALPHA - 5 * BETA - 1
      IF (POLC .EQ. 0) GOTO 40
      IF (POLALT .GE. 0) GOTO 70
      GOTO 60
40    IF (POLC .GE. POLALT) GOTO 50
50    K74R = 0
      RETURN
60    K74R = 1
      RETURN
70    K74R = 1 - K74R
      RETURN
80    IF (JS4 .EQ. 0) GOTO 15
      IF (JS11 .EQ. 1) GOTO 15
      IF (JT .LE. 10) GOTO 15
      IF (JT .GE. 26) GOTO 15
      IF (JSW .GE. 26) GOTO 15
      JR = 1
      GOTO 9
      END

```

c K74RXX not used. Only next line is changed from version

c used before 7/23 15:11

FUNCTION K74RXX(J,M,K,L,R,JA)

C BY EDWARD FRIEDLAND

C TYPED BY JM 3/20/79


```

c k74dummy added by ax 7/22/93
  K74R=JA      ! Added 7/32/93 to report own old value
  IF (M .NE. 1) GOTO 9
  ALPHA = 1.0
  BETA = .3
  IOLD = 0
  QCA = 0
  QNA = 0
  QCB = 0
  QNB = 0
  K74R = 0
  k74dummy=0
  JSW = 0
  JS4 = 0
  JS11 = 0
  JR = 0
  JL = 0
  JT = 0
  JSM = 1
9   IF (JR .NE. 1) GOTO 10
  K74R = 1
  k74dummy=1
  RETURN
10  IF (M .LE. 2) GOTO 30
  IF (IOLD .EQ. 1) GOTO 20
  IF (J .EQ. 0) QCA = QCA + 1
  QNA = QNA + 1
  ALPHA = QCA / QNA
  QCA = QCA * .8
  QNA = QNA * .8
  GOTO 30
20  IF (J .EQ. 0) QCB = QCB + 1
  QNB = QNB + 1
  BETA = QCB / QNB
  QCB = QCB * .8
  QNB = QNB * .8
30  IOLD = K74dummy
C CHECK FOR RANDOM
  IF (M .EQ. 37) GOTO 80
  IF (M .GT. 37) GOTO 15
  IF (M .EQ. 1) GOTO 15
  IF (J .EQ. JL) JSM = JSM + 1
  IF (JSM .GE. 3) JS4 = 1
  IF (JSM .GE. 11) JS11 = 1
  IF (J .NE. JL) JSW = JSW + 1
  JSM = 1
  JT = JT + J
15  POLC = 6 * ALPHA - 8 * BETA - 2
  POLALT = 4 * ALPHA - 5 * BETA - 1
  IF (POLC .EQ. 0) GOTO 40
  IF (POLALT .GE. 0) GOTO 70
  GOTO 60
40  IF (POLC .GE. POLALT) GOTO 50
50  K74R = 0
  k74dummy = 0
  RETURN
60  K74R = 1

```

```

k74dummy=1
  RETURN
c70   K74R = 1 - K74R
70   K74R = 1-k74dummy
  RETURN
80   IF (JS4 .EQ. 0) GOTO 15
      IF (JS11 .EQ. 1) GOTO 15
      IF (JT .LE. 10) GOTO 15
      IF (JT .GE. 26) GOTO 15
      IF (JSW .GE. 26) GOTO 15
      JR = 1
      GOTO 9
  END
  FUNCTION K93R(J,M,K,L,R,JA)
  K93R=JA      ! Added 7/32/93 to report own old value
  K93R=1
  IF(R.LT..5) K93R=0
  RETURN
  END
  FUNCTION K36R(J,M,K,L,R,JA)
C BY ROGER HOTZ
C TYPED BY JM
C EDITED FROM BASIC BY AX, 2/11/79
  K36R=JA      ! Added 7/32/93 to report own old value
  K36R = 1
  IF (M .GE. 1 .AND. M .LT. 100) PR0BC = .1
  IF (M .GE. 100 .AND. M .LT. 200) PR0BC = .05
  IF (M .GE. 200 .AND. M .LT. 300) PR0BC = .15
  IF (M .GE. 300) PR0BC = .0
  IF (R .LT. PR0BC) K36R = 0
  RETURN
  END
  Integer FUNCTION GRASR(JPICK, MOVEN, ISCOR, JSCOR, RANDO,JA)
  DIMENSION NMOV(4)
  GRASR=JA      ! Added 7/32/93 to report own old value
c Next line for debugging
c   if(moven. eq. 57) write(6,99) jscor
c99 format(' TEST from GRASR at move 57. jscor = ', i6)
  IF (MOVEN .NE. 1) GO TO 9997
  DO 9996 I = 1, 4
  NMOV(I) = 0
9996   CONTINUE
  NMOVE = 0
  IGAME = 0
  N = 0
9997   CONTINUE
  IF (MOVEN - 1) 25, 25, 26
25   GRASR = 0
  RETURN
26   IF (MOVEN - 51) 1, 2, 3
1   GRASR = JPICK
  RETURN
2   GRASR = 1
  RETURN
3   IF (MOVEN - 57) 4, 5, 6
4   IF (MOVEN - 52) 9, 9, 10
10  NMOV(MOVEN - 52) = MMOVE + JPICK

```

```

9  GRASR = JPICK
   IF (GRASR -1) 7, 8, 8
7  MMOVE = 2
   GO TO 11
8  MMOVE = 4
11 RETURN
5  IF (JSCOR - 135) 19, 19, 20
20 J = NMOV(2)
   GO TO (12, 12, 30, 31, 32), J
31 IF (NMOV(1) - 3) 12, 35, 12
35 IF (NMOV(3) - 3) 12, 16, 12
32 IF (NMOV(1) - 5) 12, 33, 12
33 IF (NMOV(3) - 5) 12, 16, 12
30 IF (NMOV(1) - 2) 12, 34, 12
34 IF (NMOV(3) - 4) 12, 40, 12
40 IF (NMOV(4) - 2) 12, 41, 12
12 IGAME = 1
   N = RANDO * 10.0 + 5.0
   GRASR = 0
   RETURN
16  IGAME = 2
   GRASR = 0
   RETURN
19  IGAME = 3
27  GRASR = 1
   RETURN
41  IGAME = 4
42  GRASR = 0
   IF (MOVEN - 118) 44, 43, 43
43  IGAME=2
44  RETURN
6  GO TO (21, 22, 27, 42), IGAME
21 IF (N) 23, 23, 24
23 GRASR = 1
   N = RANDO * 10.0 + 5.0
   RETURN
24 GRASR = JPICK
   N = N-1
   RETURN
22 GRASR = JPICK
   RETURN
   END

```