```
JOBHUNTER 2/10/72 (JBH,30)
/WITH ←N


/VARIABLES
JBUFP=DTEM                  /6 TABLE PTRS.
/DTEM+6   /RESERVED FOR EXPANSION
JQN=DTEM+7                  /CURRENT QUESTION NUMBER
JBF=DTEM+11                 /BRIEF MODE FLAG
JANS=DTEM+12                /CORE ANSWER PTR
JDANS=DTEM+13               /DRUM ANSWER PTR
JDTR=DTEM+14                /DRUM TEXT RELOCATION FACTOR
JTTOCP=DTEM+15              /TEXT TOC PTR
JQTOCP=DTEM+16              /QN TOC PTR
JTBUFP=DTEM+17              /TEXT BUF PTR
JQBUFP=DTEM+20              /QNBUF PTR- ALSO REFERENCED AS CJQBP
JTBUFC=DTEM+21              /TEXT BUF CURRENT PTR
JQBUFC=DTEM+22              /QN BUF CURRENT PTR
JOJM=DTEM+23                /OLD JMODE & TEM STORAGE
JLGQN=DTEM+24               /LOCATION OF LAST GQN; EOM FOR NULLING
JLEVEL=DTEM+25              /LEVEL PTR INTO JQN REGISTERS
JLQNT=DTEM+26               /LAST QN TYPED
JORG=DTEM+30                /LOC OF USER'S INIT; NUL MODE IN INS PART
                            /ALSO REFERENCED AS CJORG
JHTSU=DTEM+31               /TRAP STARTUP LOCATION
.REASK=IOT I 5064           /IOT PUT IN TTTSU


/QN TABLE ENTRY - 4 WORDS CREATED AT GQN:
/ QN WORD 1
/ QN WORD 2
/ BITS AND "QUESTION POINTER" (POINTER TO BLOCK OF TEXT PTRS)
/ -0 OR ANSWER TEXT MOBY POINTER
/BITS: 0,RECONS; 1,LIST START;2,STOP


/JMODE:
/-3 NULL
/-1 SIMULATE
/0  NORMAL  (EXCLUDES KILLED QUESTIONS)
/1  TOC ( = TYPE OUT FOR CORRECTION = RECONSIDER)
/2  LIST STOP
/3  LIST CONTINUE
/4  DEAD (QUESTION ANSWERED THEN KILLED. ACTS LIKE 0)
↑L
```

```
/SEARCH SUBROUTINE; QN TO BE SEARCHED FOR IN B,C

SEARCH,   CLL↑UCML        /SET FOR READTQ
          DAP H
          LAC I (JQBUFC) /START WITH SEG NOW IN CORE
          AND (7777)
          DAC D           /FOR ENDTEST
          DAC F

SEAR8,    LAW 2           /BEGIN SEARCHING A SEG
          ADD I (JBUFP+3)          /PTR TO SEARCH WITH
SEAR2,    DAC E
          SAS I CJQBP
          JMP SEAR6       /DIFFERENT FROM TOP OF LAST SEG
          LAW 7777        /SAME
          AND I (JQBUFC)
          SAD I (JQTOCP)
          JMP SEAR1       /WE'RE IN LAST SEG, START AT BEGINNING
          LAC E
SEAR6,    SAD I (JBUFP+4)          /CHECK FOR TOP OF ANY SEG
          JMP SEAR7       /GET NEXT SEG
SEAR3,    LAC I E
          SAD B           /CHECK FIRST WORD OF QN
          JMP SEAR4
          LAW 4
SEAR5,    ADD E
          JMP SEAR2
SEAR4,    IDX E
          LAC I E
          SAD C           /CHECK SECOND WORD
          JMP SEARF
          LAW 3
          JMP SEAR5
SEARF,    IDX H           /2ND RETURN
          IDX E
          JMP HEXIT

SEAR1,    LAC I (JBUFP+4)          /START AT BEGINNING OF SEGS
          JMP SEAR9
SEAR7,    IDX F           /GET NEXT SEG
SEAR9,    SAD D           /DID WE START HERE?
          JMP HEXIT       /YES, GIVE RETURN 1
          DAC F           /SETUP PTR TO DA
          LAW SEAR8       /CALL READTQ; RETURN TO SEAR8
↑L
```

```
/ROUTINE TO READ TEXT SEG OR QN SEG (LINK=1 FOR QN)
READTQ,   DAP A
          LAC I (JTBUFC)
          SZL
          LAC I (JQBUFC)
          LIA             /FIND OUT WHAT'S IN CORE
          AND (7777)
          DIP F           /COMPARE ADR PARTS ONLY
          SAD F           /SAME AS WHAT'S WANTED?
          JMP AEXIT       /YES, RETURN
          SPI I
          JMP READ1       /VALID ON DRUM, DON'T WRITE OUT
          DAC G           /WRITE OUT
          LIO I G         /DRA
          LAC I (JBUFP)
          SZL
          LAC I (JBUFP+3)           /CORE ADR
          SNI             /NO DRA?
          JMP READWN      /WRITE NON-ADR
          WAI+22          /REWRITE
           JSP JIOPER
          JMP READ1       /NOW GO READ
READWN,   WNIH+2          /WRITE OUT
           JSP JIOPER
          DIO I G         /SAVE DRA

READ1,    LAC F           /READ,
          SZL I
          JMP READ1T
          DAC I (JQBUFC) /UPDATE CURRENT POINTER
READ2T,   LIO I F         /GET DRA
          SNI
          JMP AEXIT       /NO DRA, NOTHING IS DESIRED
          LAC I (JBUFP)
          SZL
          LAC I (JBUFP+3)           /GET CORE ADR
          RAI+2           /READ
           JSP JIOPER
          JMP AEXIT

READ1T,   DAC I (JTBUFC) /UPDATE CURRENT PTR
          SUB I (JBUFP+1)           /CALCULATE RELOCATION VALUE
          DAC G
          LAW I 2
          ADD I (JBUFP+1)
          SUB I (JBUFP)
          MUL G
          DIV B17
C65,       65             /CONSTANT 65
          SUB I (JBUFP)
          SUB B16
          DAC I (JDTR)    /SETUP RELOCATION REGISTER
          JMP READ2T
↑L
```

```
/TELETYPE SUBROUTINES, CLOBBER H.

/SPACE
SPAC,     LIO (SPAT)

/TOS FROM IO
JTOS,     DAP H
          LAI
          TOS
           JMP JTTER
          JMP HEXIT

SPAT:     TEXT / #/

/TT ERROR ROUTINE AND .REASK=SICKTT IOT
REASK.,
JTTER,    LAW 2
          SAS I (ERCODE) /INTERUPTED BY NULL?
          EOTHNG         /NO, HANG
          LIO (JINTT)    /YES, COMMENT AND REASK
          JSP JTOS
          LIO C64        /(64)
          LAC I (JMODE)  /BUT IF LISTING,
          AND (777776)
          SAD B16
          LIO B15        /(4) JUST RESTART
          JMP KILLFI     /GO REASK
JINTT:    TEXT /
INTERRUPTED.#/

/ERROR PRINTOUT ROUTINE
JTERR,    AND (177777)
          TOS
           JMP JTTER
          JMP TYIBM1     /GO HALT USER ON P 28 OR SO

/IOP ERROR ROUTINE
JIOPER,   LIA
          LAC (JIOPRT)
          TOS
C64,       64            /CONSTANT 64 IN TT ERROR RETURN
          Ø
JIOPRT:   TEXT /
JBH FASTR ERROR.#/


/SUBR TO COPY QN INTO B,C
GQUES,    DAP H
          LAC I (JQN)
          DAC B
          LIO I (JQN+1)
          DIO C
          JMP HEXIT
↑L
```

```
/INIT IOT
INIT,     DZM I (JLQNT)
          LAC (.REASK)
          DAC I (TTTSU)  /FOR THE BENEFIT OF THOSE WHO TOS OR TIS
          LAW 2
          ADD I (JBUFP)
          DAC I (JTBUFP) /WHERE TO PUT TEXT
          CMA
          DAC I (JDTR)
          LAC I (JBUFP+1)
          DAC I (JTTOCP) /IN WHICH SEG TO PUT TEXT
          DAC A
          DZM I A        /DRA OF SEG
          DAC I (JTBUFC) /WHAT'S IN CORE
          ADD (JMP-13)
          DAC I (TISMAX) /FIR TISMAX RECOVERY
          LAW 2
          ADD I (JBUFP+3)
          DAC I CJQBP    /WHERE TO PUT QN ENTRIES
          LAC I (JBUFP+4)
          DAC I (JQTOCP) /IN WHICH SEG TO PUT QN ENTRIES
          DAC A
          DZM I A        /DRA OF SEG
          DAC I (JQBUFC) /WHAT'S IN CORE
          IDX I (USERPC)
          DAC I CJORG    /WHERE TO RESTART
          CLA            /IN NORMAL MODE
          JMP INIT1      /RESTART
↑L
```

```
/KILL & RESTART IOT
GTYOQF,   LAC B           /FULL FAIL ENTRY FROM GTYOQ IOT
          DAC I (JQN)     /RESTORE QN
          DIO I (JQN+1)
          LIO (74)        /BITS FOR FULL FAIL
KILLFI,   LFI             /FALL INTO KILL IOT. ENTRY FROM ~ROUTINES
/IOT ENTRY
KILL,     SZF 2
          JMP KILL1
          SZF I 1         /PF 1, NOT PF2, RECONSIDER
          JMP RESTART     /PF1&2 CLEAR, RESTART ONLY

KILL12,   SZF 3           /GET QN TO KILL OR RECONS, WHERE?
          JMP KILLG1      /AC,IO
          IDX I (USERPC)  /CALLING SEQUENCE
          DAC C
          LIO I C
          IDX I (USERPC)
          DAC C
          LAC I C
          JMP KILLG2

KILLG1,   LIO I (USERAC)  /GET QN FROM AC,IO
          LAC I (USERIO)
KILLG2,   DIO B
          DAC C

KILL6,    JSP SEARCH      /GET QN TABLE SEG IN CORE, POINT E AT QUES PTR
           JMP RESTART     /JUST RESTART IF NO SUCH NUMBER
          LAC I (JMODE)
          SZF I 2
          JMP KILL11      /RECONS ON CLEAR PF2 (PF1 SET IF HERE)
          SZF 1           /KILL ON CLEAR PF1 (2 SET)
          SAS (-1)        /FAIL, REASKS GET HERE, RECONS IF SIMULATING
          JMP KILL10
KILL11,   LAC I E         /RECONSIDER
          IOR B0
          DAC I E
          SKP I
KILL10,   DZM I E         /KILL
CHARLY,   LAW I 0         /CLC
          DIP I (JQBUFC)  /MARK BUFFER MODIFIED
```

```
RESTART,  LAW I 3
          SZF I 4
          JMP REST6
          SAD I (JMODE)
          DZM I (JMODE)   /FLUSH NULL MODE
          SZF 5           /RESTART WHERE?
          JMP REST1       /FROM GQN OR CALLING SEQ
          LAW 3           /FROM INIT
          SZF I 6         /DETERMINE MODE TO REST IN (3 FOR LIST IN AC)
          LAW I 1         /SIM MODE
INIT1,    DAC I (JMODE)
          LAC (LAC JQN)   /RESTORE QN TO Ø,Ø & LEVEL
          DAC I (JLEVEL)
          DZM I (JQN)
          DZM I (JQN+1)
          LAC I CJORG     /GO BACK TO INIT
          JMP GO
↑L
```

```
KILL1,    SZF I 1          /COME HERE IF PF2 SET
          JMP KILL12       /KILL FROM AC,IO OR CALLING SEQ
          SZF I 3          /CURRENT QN; TYPE FIX FIRST?
          JMP KILL5
          LAC I (JMODE)    /FAIL IOT: TYPE "FIX" OR "FIX:"
          LIO (JTXFIX)
          SPA              /DEPENDING ON JMODE
KILL13,   LIO (JTXFXC)     /ENTRY FOR UNANS Q, NOT NULL MODE
          SAS (-3)         /DON'T TYPE IN NULL MODE
          JSP JTOS         /TYPE-OUT ROUTINE. PTR IN IO.
KILL5,    JSP GQUES        /SETUP CURRENT QN IN B,C
          JMP KILL6

JTXFIX:   TEXT / FIX#/
JTXFXC:   TEXT /
FIX:#/



REST6,    SZF I 5
          JMP R1           /NO RESTART ON RESTAR+ 0,1
          DAC I (JMODE)    /RESTAR + 2,3 ENTER NULL MODE
REST1,    LAC I (JLGQN)
          SZF 6
          JMP REST3        /RESTART AT BEGINNING OF JOB
          JSP TRACE        /RESTART FROM CALLING SEQUENCE
          IDX I (USERPC)
          DAC C            /QN(1)
          LAC I C
          DAC I (JQN)
          IDX C            /QN(2)
          LAC I C
          DAC I (JQN+1)
          LAC A            /WHERE TO GO
REST3,    AND (7777)       /GET RID OF EOM IN JLGQN
          DAC I (USERPC)
          DAC C
          LAW 77
          AND I C
          LIA
          LAW JQN          /SET UP LEVEL (FOR RESTART FROM CALL)
          DAC C
          IDC↑ULFI
          DAC D
REST2,    LCH I C
          LCH I D
          SZA              /SEARCH FOR 0
          JMP REST2
          LAC C            /POINTS TO LAST NON-0 BYTE (OR FIRST ONE)
          DAC I (JLEVEL)
          JMP GTYOQ9       /GO TO GQN
↑L
```

```
/GTYOQ IOT.  UPLV, DNLV, XDNLV PART:
GTYOQ,    JSP GQUES       /SAVE OLD QN. NOTE THAT 2ND WD IN IO.
          LAC I (JLEVEL)
          DAC A           /SETUP LEVEL PTR IN A
          SZF I 2         /SET FOR UPLV, DNLV
          JMP GTYOQ5      /CLEAR FOR NONE, XDNLV
          SZF I 1         /SET FOR XDNLV, DNLV
          JMP GTYOQ6      /CLEAR FOR UPLV, NONE
GTYOQ7,   CLF 1           /FOR NEXT TIME AROUND (XDNLV))
          LAC A
GTYOQ8,   IDC             /DNLV (OR END OF UPLV)
          SAS (JMP JQN-1)          /ERROR ON UPLV
          SAD (LAC JQN+2)          /ERROR ON DNLV
          JMP GTYOQF
          DAC A
          DAC I (JLEVEL)

/GQN PART; SOME ALSO USED BY XDNLV
GTYOQ5,   SZF 1           /FOR XDNLV, GO INCR ON CURRENT LEVEL
          JMP GTYQ51
          SZF I 3         /GQN BIT?
          JMP TYOQ
GTYQ51,   LCH A           /INCR QN
          ADD B5
          SAD B17
          JMP GTYOQF      /OVERFLOW
CHLCM,    DCH A
          SZF 1
          JMP GTYOQ7      /XDNLV
GTYOQ9,   LAC I (USERPC) /SET LOCATION OF LAST GQN
          IOR ML4         /AN EOM TO TYPE IN FOR NULLING
          DAC I (JLGQN)
          LAC I (JMODE)   /SAVE MODE BEFORE GENERATING NEW MODE
          DAC I (JOJM)    /REFERENCED IN TYOQ
          JSP GQUES       /COPY QN INTO B,C
          JSP SEARCH      /LOOKUP CURRENT QN
          JMP QNEWG       /GO MAKE NEW ENTRY
GTYOQ4,   JSP TRACE       /GET USER'S QUESTION POINTER
          LAW 7777
          AND A
          DAP I (JOJM)    /SAVE IN JOJM
          LIO I E         /OLD QUES PTR TO IO
          DAC I E         /NEW ONE TO TABLE
          XAI             /TEST IDENTITY OF ADDRESSES
          AND (7777)
          SZF 3
          SZA
          JMP QNOT        /DIFFERENT PLACE OR NEW QUESTION
↑L
```

```
/SET UP MODE FOR PREVIOUSLY ASKED QUES - SIM, LIST, OR TOC
GMODE,    LAW I 7777
          NAI
          SZA I           /ANY BITS SET ON QUES PTR FROM TABLE?
          JMP GMOD3
          CLC
          DIP I (JQBUFC) /YES, MARK BUFFER MODIFIED
GMOD3,    LAW 1
          SPI             /RECONSIDER BIT SET?
          JMP GMOD4
          RIL 1S
          LAW 3
          SAS I (JMODE)   /IN LIST MODE?
          SPI             /OR LIST START BIT?
          JMP GMOD2       /YES, ENTER LIST MODE & TEST STOP BIT
          LAW I 1         /NOT IN L MODE, NO L START OR RECONS BIT,
                          /SO ENTER SIM MODE
GMOD4,    DAC I (JMODE)   /SET MODE
          IDX E
.TYOQ,    LAC I E         /NEW QN COME HERE
          DAC I (JDANS)   /MOBY PTR TO PREVIOUS ANS OR -0
↑L
```

```
/TYOQ PORTION OF GTYOQ IOT
TYOQ,     LAC I (JMODE)
          SZF 4           /TYOQ BIT OF IOT?
          SPA             /TEST FOR SIM OR NULL MODE
          JMP R1          /NO TYPE
          LAC I (JOJM)
          SPA             /WAS PREVIOUS QUESTION IN SIM. MODE?
TYOQ1,    CLF 7           /YES, CLEAR FLAGS 1&2
          LAC (760000)
          SZF 5           /SUPPRESS CR BIT
          JMP TYOQ13
          TYO             /TYPE CR
           JMP JTTER      /TYO ERROR
TYOQ13,   SZF 6           /SUPPRESS QN BIT
          JMP TYOQ5B
          SZF 5           /SUPPRESS INDENT (CR)
          JMP TYOQ3
          LAC (LAC JQN)   /BEGIN ROUTINE TO INDENT
          DAC F
TYOQ2,    LCH I F
          SZA I
          JMP TYOQ3       /FINISHED INDENTING, GO TYPE QN
          JSP SPAC        /TYPE A SPACE
          JMP TYOQ2

TYOQ3,    LAW JQN
          DAC F
          LIO (11)        /FLAGS 3&6 FOR SNM
          LFI
          LCH I F
          JMP TYOQ12      /SO INITIAL 0 PRINTS

TYOQ9,    SUB B5          /ALPHA PRINT #
          RCL 7S          /IO WAS CLEAR
          DIV C26.        /LETTERS IN ALPHABET
C26.,      26.
          IDA
          CMA             /SETUP FOR ISP
          DAC H
          LAW CHARAC RA
          AAI
          RCR 6S          /CHAR INTO TOP(IO)
TYOQ10,   LAI
          TYO
           JMP JTTER
          ISP H
          JMP TYOQ10      /RE-TYPE LETTER
TYOQ4,    LCH I F
          SZA I
          JMP TYOQ5       /FINISHED, GO TYPE QUESTION
TYOQ12,   CML↑USCI↑USZL   /ENTRY. CHANGE & TEST ALPHA FLAG
          JMP TYOQ9       /IO CLEAR
          RCL 6S          /# TO BE TYPED TO IO
          LAW TYOQ4
          JMP .SNMJ       /CALL SNM WITH RETURN TO TYOQ4
TYOQ5,    JSP SPAC        /SPACE AFTER QN
↑L
```

```
                        /ROUTINE TO TYPE QUESTION
TYOQ5B,   LAC I (JQN)   /DOES THIS QN = THE LAST 1 TYPED?
          STF 1         /SIGNIFIES NOTHING MODE
          LIO I (JBF)
          SAS I (JLQNT)
          JMP TYOQ5A
          LAC I (JQN+1)
          SAS I (JLQNT+1)
TYOQ5A,   SNI
          JMP TYOQ8     /IF QN= LAST QN TYPED, OR JBF=Ø
          LAW 7777
          AND I (JOJM)
          DAC A         /PTR TO SHORT MODE PTR
          LAW 1
          SAS I (JBF)   /UNLESS SHORT MODE,
          IDX A         /MAKE I LONG
          LIO I A
          JSP JTOS      /TYPE QUESTION

          CLF 1         /MEANING NOT IN NOTHING MODE
TYOQ8,    JSP SPAC      /COME HERE FOR NOTHING QUESTION, PF1 SET
          LAC I (JQN)   /MARK QN AS LAST 1 TYPED
          DAC I (JLQNT)
          LAC I (JQN+1)
          DAC I (JLQNT+1)
          LAW 3
          SZF I 1       /IF TYPING NOTHING QUESTION,
          SAS I (JBF)   /OR NOT IN HOW MODE,
          JMP R1        /DONE
          IDX A
          LIO I A
          JSP JTOS      /TYPE HOW MODE
          JMP TYOQ1     /GO TYPE NOTHING QUESTION


/REMOTE ROUTINE FOR MODE SETTING
GMOD2,    RIL 1S        /3 IN AC FOR LIST CONTINUE MODE
          SPI           /IF LIST STOP BIT,
          LAW 2         /MAKE IT LIST STOP MODE
          JMP GMOD4

/REMOTE ROUTINE TO DO UPLV
GTYOQ6,   CLA
          DCH A         /CLEAR VALUE AT OLD LEVEL
          LAW I 1
          ADD A         /UNSTEP A
          IDC
          JMP GTYOQ8
↑L
```

```
/ROUTINE TO MAKE NEW QN TABLE ENTRY
/QN FROM B,C; LINK MUST BE SET

QNEWG,    LAC I CJQBP
          SAS I (JBUFP+4)          /AT END OF BUFFER?
          JMP QNEW1       /NO
          IDX I (JQTOCP) /NEW SEGMENT
          SAD I (JBUFP+5)          /OUT OF ROOM?
          JMP QNEWBF      /TYPE ERROR & TRAP
          DAC E
          DZM I E         /NO DRA
          LAW 2
          ADD I (JBUFP+3)          /NEW CORE ADR FOR TABLE
QNEW1,    DAC E           /PTR TO BEG OF 4 WD TABLE
          ADD B15         /(4), NEXT TABLE
          DAC I CJQBP
          LAC I (JQTOCP) /CURRENT SEG PTR
          DAC F
          JSP READTQ      /INTO CORE (LINK LEFT SET)
          LAC B           /QN TO TABLE
          DAC I E
          IDX E
          LAC C
          DAC I E
          IDX E
          CLF 3           /MEANING NEW QUESTION
          JMP GTYOQ4

QNEWBF,   JSP JTERR
          TEXT /
TOO MANY QUESTIONS.
/
ML4,      740000


/STUFF FOR QUES NOT ASKED BEFORE OR KILLED
/OLD QUES PTR IS IN IO

QNOT,     IDX E
          LAW I 3
          SAD I (JMODE)  /IF NULLING, CONTINUE
          JMP QMOD2
          LAC I E         /OLD ANS PTR
          SAS (-0)        /IF ANSWERED BEFORE
          SNI↑USZF I 3    /AND ASKED AND KILLED,
          JMP QMOD1
          LAW 4           /THEN ENTER DEAD MODE
          SKP I
QMOD1,    CLA             /ENTER NORMAL MODE
          DAC I (JMODE)
QMOD2,    CLC
          DAC I E         /-0 ANS PTR
          DIP I (JQBUFC) /MARK BUFFER CHANGED
          JMP .TYOQ
↑L
```

```
/IOT TO CONVERT MOBY PTR INTO CORE PTR & GET INTO CORE
GTEXT,    LAC I (USERAC)
          DAC D
          LAW R1
/SUBR ENTRY
.GTEXT,   DAP H
          LAW I 2
          SUB I (JBUFP)
          ADD I (JBUFP+1)          /CALCULATE BUFFER LENGTH
          DAC F
          LAC D          /GET MOBY PTR
          CLL↑USCI       /CLL FOR READTQ
          RCL 2S
          RAR 1S         /REMOVE TOP 2 BITS
          CLI↑USWP       /POSITION FOR DIVIDE
          DIV F
C67,       67            /CONSTANT 67
          ADD I (JBUFP+1)          /ADD BASE OF TOC TO QUOTIENT
          DAC F          /SET UP FOR READTQ
          JSP READTQ     /GET THIS BUFFER INTO CORE
          LAC D
          SUB I (JDTR)
          DAC I (FSA)    /SAVE IN FSA
          DAC D
          JMP HEXIT
↑L
```

```
/TYINV IOT
TYINV,     LAC I (JTBUFP)
           SUB I (TISMAX) /SET 13 WORDS BELOW END OF BUFFER BY INIT IOT
           RAL 2S         /BYTE PART OF PTR OUT OF THE WAY
           SMA            /IS PRESENT TEXT SEG FULL ENOUGH
           JSP JNTS       /YES, START ANOTHER
           LAC I (JTBUFP)
           DAC I (JOJM)   /CURRENT PTR TO END OF TEXT
           JSP GQUES      /GET CURRENT QN IN B,C
           JSP SEARCH
            C16RET+1      /ILLEGAL SEQUENCE OF IOTS
           LAC I (JQBUFC) /Q BUF SEG
           DAC I (ATEM+5) /SEE TYIBC+24
           IDX E          /GET PTR TO ANSWER
           STF 1          /FOR .EDITJ

/CHECK JMODE, TYPE OLD ANS AND THINGS IF NECESSARY
TYINV1,    SZF 4
           JMP TYINN      /FLAG 4 FORCES TYPEIN IN ALL CASES
           LAC I (JMODE)
           SAS (-3)
           JMP TYINV3
           LAW JLGQN      /NULL MODE, GO TYPE EOM
           STF 6          /IN FROM CORE
           JMP TYINNN

TYINV3,    AND (3)        /AND WITH 3 TO MAKE 4 LOOK LIKE 1
           SZF 6
           AND B16        /GET RID OF TOC MODE
           SZA I
           JMP TYINN      /NORMAL MODE; TOC MODE IF TYPEIN FROM CORE
           LAC I E        /ANSWER POINTER
           SAD (-Ø)
           JMP TYINVK     /NOT ANS BUT NOT NORMAL - FAIL.
           DAC I (JDANS)
           DAC D
           JSP .GTEXT     /RETRIEVE ANSWER
           LIO D
           DIO I (JANS)

TYINV2,    LAC I (JMODE)
           SZF I 6
           SPA
           JMP TYIV       /DONE HERE IF IN SIM MODE OR FROM CORE
                          /LIST, TOC GET HERE
           JSP JTOS       /TYPE OLD ANSWER
           LAW 1
           SAS I (JMODE)
           JMP TYIV       /DONE IN LIST MODE
           JSP SPAC       /TOC MODE; SPACE
                          /FALL THROUGH TO GET NEW ANSWER
↑L
```

```
/TYPE-IN SECTION  FROM CORE ON PF6
/LEAVES JTBUFP POINTING TO BEG OF TEXT
/       JOJM TO END
/TRANS TO USERAC

/STORAGE
/A  SUBR RETS
/B  DCH I PTR FOR TEXT
/C  PTR TO BEG TEXT; PTR FOR COPY-BACK
/D  TEXT PTR ON PF6; OW TRANS
/E SAVED IN ATEM+4 (PTR TO ANS PTR)
/F  ARG FOR READTQ
/G  SUBRS USE

TYINN,    LAC I (USERAC) /FOR TYPE IN FROM CORE
TYINNN,   DAC D
          CLL
          LAC I (JTTOCP)
          DAC F
          JSP READTQ     /GET TEXT SEG BEING FILLED
          LAC I (JTBUFP)
          DAC B          /DCH I POINTER FOR TEXT TYPED IN
JRBTY1,   DAC C          /PTR TO BEGINNING FOR COPYBACK
          SZF 6
          JMP JTYIEL     /FROM CORE
          LAC B
          TIS
           JMP JTYIER    /ERROR; IF TISMAX, START NEW BUF

JRBTY5,   LAC E          /SAVE PTR TO CURRENT ANS PTR
          DAC I (ATEM+4) /THROUGH EDIT AND ←C
          SZF 6
          JMP JRBTY4
          LIO I (JTBUFP) /TEXT PTR FOR EDIT
          JSP .EDITJ
           JMP TYINNR    /RUBOUT, TYPE #, REASK
JRBTY4,   DIO I (JOJM)   /PTR TO END OF TEXT
          LAC I (ATEM+4) /RESTORE PTR FOR ←.
          DAC E          /NOT NEEDED IF EDIT CHANGED
          LCH I C        /LOOK FOR ←
          RCL 6S
          LCH I C
          RCR 6S
          SAD (FLEXO ← )
          JMP TYINBA     /HANDLE ←
TYIBR,    LAC I (JTBUFP)
TYIBRB,   DAC I (FSA)
TYIBRC,   DAC I (JANS)
          ADD I (JDTR)
          DAC I (JDANS)
          LIO I (ATEM+4) /RESTORE PTR TO (OLD) ANSWER PTR
          DIO A
```

```
CHARLK,    SAS I A          /IS NEW POINTER THE SAME (AS ← AFTER RECONS)
           CLF 1
           DAC I A          /UPDATE PTR IN QN TABLE
           CLC
           SZF I 1
           DIP I (JQBUFC) /BUFFER IS CHANGED IF PTR CHANGED
↑L
```

```
/VERIFY AND FINISH UP
TYIV,     SZF I 5
          JMP TYIN        /DON'T VERIFY
          JSP TRACE       /GET SYNDEF ADDRESS IN A
          LIO A
          JSP .STVJ
           JMP TYINVF     /LOSE
TYIN,     LAC I (JOJM)
          SUB I (JTBUFP)
          CLI↑UCMI↑USWP
          SNI I
          DIP I (JTBUFC) /MARK TEXT BUF CHANGED IF PTR CHANGED
          LAC I (JOJM)   /MUST CONTAIN POINTER TO END OF TEXT
          DAC I (JTBUFP)
          JMP R1

TYINVK,   DZM I (JMODE)  /PREVENTS FIX: FIX ON UNANSWERED QUES
          STF 7
          JMP KILL13     /TYPE "FIX:" AND REASK

TYINVF,   LIO (77)       /TYPE FIX AND REASK
          JMP KILLFI

TYINNR,   LAC CHLNUM     /#
          TYO
           JMP JTTER
TYINNI,   LIO C67
          JMP KILLFI     /REASK
↑L
```

```
/TYPE IN FROM CORE AND TIS ERRORS
/NOTE THAT TISMAX IS 13 WORDS BELOW END OF BUFFER

JTYIER,   DAC B          /TIS ERROR.  SAVE POINTER
          LAW 3
          SAS I (ERCODE)
          JMP JTTER      /NOT A TISMAX ERROR
JTYIEL,   CLI            /COPY ANSWER FROM CORE AND TISMAX ERR RUTIN
          SZF I 6
          TYI 2-1        /EMPTY TT BUFFER AND DON'T HANG
          SZF 6
          LCH I D        /FROM CORE
          SPI
          JMP JTYIEH     /SHOULD HAVE HUNG. MAKE NEW BUF WHILE WE CAN
          DCH I B
          LIO B          /POINTER TO END, NEEDED IF EOM FROM CORE
          SAD (CHARAC R#)
          JMP JRBTY5     /EOM. DONE.
          LAC (JMP-1)
          ADD I (JBUFP+1)
          SAS B          /BUFFER COMPLETELY FULL?
          JMP JTYIEL     /NO, LOOP

/CREATE NEW TEXT SEG AND COPY TEXT BACK TO BEGINNING
JTYIEH,   LAC B
          DAC I (JOJM)   /SAVE POINTER TO END OF TEXT
          LAW 2
          ADD I (JBUFP)
          SAD C          /PTR TO BEG OF TEXT TYPED IN
           JMP JTYIEF    /STARTED AT BEGINNING OF BUFFER, ANS TOO LONG
          DAC B          /PTR TO BEG OF ANSWER
          JSP JNTS       /SET UP POINTERS FOR NEW SEG
          JSP READTQ     /TO WRITE CURRENT SEG ON DRUM IF NECESSARY

JTYIEM,   LCH I C
          DCH I B        /COPY TEXT DOWN TO BEGINNING OF BUFFER
          LAC C
          SAS I (JOJM)
          JMP JTYIEM
          LAC I (JTBUFP) /NEW VALUE OF PTR TO BEGINNING
          JMP JRBTY1

JTYIEF,   SZF 6          /ANSWER TOO LONG
          JMP JTYIFC
          LIO (JTYIFT)
          JSP JTOS
          JMP TYINNI
JTYIFT:   TEXT /
ANSWER TOO LONG,#/

JTYIFC,   JSP JTERR
          TEXT /
INTERNAL ANSWER TOO LONG#./

↑L
```

```
/SUBR TO SET UP PTRS FOR NEW TEXT BUF SEG
JNTS,     DAP A
          LAW 2
          ADD I (JBUFP)
          DAC I (JTBUFP) /POINTS TO BEGINNING OF TEXT AREA
          IDX I (JTTOCP) /NEXT WORD IN TOC
          DAC F
          SAD I (JBUFP 2)
           JMP JNTSE      /ALL SEGS USED
          DZM I F         /Ø DRA SINCE NEW
          JMP AEXIT
JNTSE,    JSP JTERR
TEXT /
TOO MUCH TEXT.
#/
↑L
```

```
/DECODE QUESTION NUMBER ROUTINE FOR ← INTERPRETATION
DQN,      DAP H
          DZM I (ATEM)
          DZM I (ATEM+1) /FOR BUILDING UP QN
          LAW ATEM
          DAC F          /LEVEL PTR
          DZM D          /LEVEL COUNTER, EXCLUSIVE OF INITIAL Ø
          LCH G
          SAD B1         /(CHARAC LØ)
          JMP DQN8       /HANDLE NUMBERS LIKE ØA

DQN1,     CLL            /DECODE A NUMERIC LEVEL. LINK SET AFTER A DIGIT
          DZM B
          LCH G
          SAD B1         /(CHARAC LØ)
          JMP DQNO       /CAN'T BEGIN WITH Ø AT PRESENT
DQN3,     RAL 6S
          XOR B13
          DAC C
          SUB C1Ø,       /(1Ø.)
          SMA
          JMP DQN2
          LAW 1Ø.
          MUL B
          SIR 1S
          LAC C
          AAI↑UCLL↑UCML  /ACCUMULATE VALUE IN HI BITS
          DAC B          /OVERFLOW CARRIES AROUND TO LO BITS
          LCH I G
          JMP DQN3

DQN2,     SZL I
          JMP DQNO       /NO NUMBER
          LAC B
          RAR 6S

DQN7,     DCH I F        /STORE VALUE
          SAR 6S
          SZA            /OVERFLOW CHECK
           JMP HEXIT     /R1
          IDX D          /HAVE A GOOD LEVEL
          LAC F
          SAD (LAC ATEM+7)          /NUMBER OF LEVELS CHECK
           JMP HEXIT     /R1 LOSE
          SZL I          /LINK CLEAR IF NEXT FIELD NUMERIC
          JMP DQN1
          JMP DQN6

DQN8,     LCH I F        /STEP LEVEL FOR INITIAL Ø
          LCH I G        /AND STEP CHAR PTR
↑L
```

```
DQN6,     LCH G          /DECODE ALPHA LEVEL
          DAC B
          SUB (CHARAC LZ+1)
          LIA
          ADD (CHARAC LZ+1-CHARAC LA)
          SPI↑USMA I     /MUST BE LETTER
          JMP DQNO       /ISN'T
          DZM C          /LETTER COUNTER
          JMP DQN5
DQN4,     IDX C
DQN5,     LCH I G
          SAD B
          JMP DQN4       /COUNT # OF SAME LETTERS
          LAW 26.        /CONVERT TO NUMBER
          MUL C
          RIR 7S         /VALUE FOR ALL BUT FIRST LETTER TO HI IO
          LAC B
          XOR BØ         /VALUE FOR FIRST LETTER
          AAI↑UCLL
          JMP DQN7       /GO STORE VALUE, CHECK FOR OVERFLOW

DQNO,     LAC I (ATEM)   /DONE
          DAC B          /LEAVE NUMBER IN B,C
          LAC I (ATEM+1)
          DAC C
          LAC D          /ILLEGAL IF NO LEVELS (NOT A DIGIT FIRST, OR Ø ONLY
          SZA            /R1
          IDX H
          JMP HEXIT

↑L
```

```
/SUBR TO TYPE DIAGNOSTIC ERRORS (QN AS TYPED IN)
/WITH MANY ENTRIES FOR VARIOUS ERRORS
TYIBCI,   CLL
          LAW TYIREA
TYIBIS,   LIO (TYIBIT)
TYIS,     DAP A           /MAIN ENTRY. TEXT ADR IN IO, RET IN AC
          LAC I (ATEM+2)
          DAC G
          LAC (760000)
TYISL,    TYO             /TYPE CRLF, QN
           JMP JTTER
          LCH I G
          SAD ML4
          CLF 1
          SZL             /TERMINATE ON , IF LINK SET ONLY
          SAS CHLCM
          SAD ML4         /CHARAC L#
          JMP TYISC
          SZF 3           /TERMINATE ON - IF PF3 SET
          SAS (CHARAC L-)
          JMP TYISL
TYISC,    LAC G
          DAC I (ATEM+2)
          LCH I G
          JSP JTOS        /TYPE COMMENT WHOSE ADDRESS CAME IN IO
          JMP AEXIT

TYIBIT:   TEXT / IS NOT A LEGAL QUESTION NUMBER.#/

TYIBCA,   CLL
          LAW TYIREA
TYIBAS,   LIO (TYIBAT)
          JMP TYIS
TYIBAT:   TEXT / HAS NOT BEEN ANSWERED.#/

TYIBCK,   CLL
          LAW TYIREA
TYIBKS,   LIO (TYIBKT)
          JMP TYIS
TYIBKT:   TEXT / HAS NOT BEEN ASKED.#/
↑L
```

```
/BACK-ARROW INTERPRETING ROUTINES
/DISPATCH ON CHARACTER AFTER ←
TYINBA,   CLF 4
          CLI↑UCLF 6
          LAC C
          IDC             /STEP PTR BEYOND CONTROL LETTER
          DAC G
          DAC I (ATEM+2) /SAVE CORE PTR TO ANSWER
          LCH G
          SAD CHARLC
          JMP TYIBC
          SAD CHARLH
          JMP TYIBH
          SAD (CHARAC L/)
          JMP TYIBS
          SAD (CHARAC L-)
          JMP TYIBM
          SAD ML4         /CHARAC L#
          JMP TYIVOK
          SAD (CHARAC LN)
          JMP TYIBN
          SAD CHARLL
          JMP TYIBL
          SAD (CHARAC LR)
          JMP TYINB8
          SAD CHARLK
          JMP TYINB6
          LIO C           /←NUMBER WITHOUT CONTROL LETTER, UNSTEP PTR
          DIO I (ATEM+2)
          JMP TYINB9      /TREAT AS ←R
TYINB6,   STF 6           /PF6 FOR KILLING
TYINB8,   LCH I G
          JMP TYINB9
↑L
```

```
/←L(LIST) AND ←#,←R,←K (RECONS AND KILL) INTERPRETING ROUTINE
/ACCEPT A LIST OF NUMBERS SEPERATED BY , AND - ;  MARK QN TABLE ENTRIES

/FLAGS: 1, CLEARED IF EOM SEEN; 2, CHANGE MADE, FULL RESTART;
/3, ←L NOT ←# ETC. (AFFECTS TYIS)
/  4, RESTART IN LIST MODE; 5, HAVE SEEN A - AND MAYBE A # SINCE , ;
/  6, ←K NOT ←R
/STORAGE:
/  ATEM+2: CORE PTR TO BEG CURRENT QN IN TEXT BEING INTERP
/    +3: PTR TO END OF SAME
/    +6: QN TOC PTR FOR QUESTION WHICH IS TO GET START BIT
/    +7: CORE POINTER TO QUESTION TEXT PTR IN SAME
/  E: PTR INTO QN TABLE, PARTICULARLY TO ENTRY TO GET STOP BIT
/  G: BYTE PTR OVER TEXT BEING INTERPRETED


TYIBL5,    STF 4            /COME BACK HERE ON INITIAL COMMA
TYIBL,     STF 3            /ENTRY FOR ←L
           LCH I G
           SAD CHLCM
           JMP TYIBL5
           SAD ML4
           JMP TYIBLU       /EOM, GO RESTART
           SAD (CHARAC L-)
           JMP TYIBL8
TYINB9,    RAL 6S           /ENTRY FOR ←#, ←R, ←K
           XOR B13          /DIGIT?
           SUB C1Ø.         /(1Ø.)
           SMA
           JMP TYIBR        /NO, FALL THROUGH TO USER
TYIBL6,    CLF 5            /ERRORS COME BACK HERE
           JMP TYIBL7

/MAIN LOOP
TYIBLE,    STF 4            /COME HERE ON - AT BEGINNING OF "ELEMENT"
TYIBLJ,    LIO G            /AND HERE ON - AFTER #
           STF 5
           LCH I G
           SAD CHLCM
           JMP TYIBLF
           SAS ML4
           JMP TYIBL4       /LOOK FOR ANOTHER NUMBER (PF5 SET)
           CLF 1
TYIBLF,    DZM E            /NO QUES IS TO BE MARKED STOP
TYIBLL,    LAC G
           DAC I (ATEM+3) /SAVE THRU READTQ
```

```
TYIBL9,   CLA↑UCLF 5      /LOOP ENDS WITH FOLLOWING
          SAD E
          JMP TYIBL1      /Ø, NO QUES TO PUT STOP BIT ON
          LAC BØ          /RECONS BIT
          SZF 3
          LAC B2          /LIST STOP BIT
          IOR I E
          SZF I 3
          SZF I 6
          SKP I
          CLA             /KILL ON PF6 AND NOT PF3
          DAC I E         /SET ON THIS QUES
          CLC↑USTF 2
          DIP I (JQBUFC)
↑L
```

```
TYIBL1,   LAC I (ATEM+6) /MARK START QUES IF ANY
          SZA I
          JMP TYIBL2
          DAC F
          JSP READTQ     /GET QN SEG INTO CORE
          LAC I (ATEM+7)
          DAC E
          LAC B1         /MARK IT LIST START
          IOR I E
          DAC I E
          CLC↑USTF 2
          DIP I (JQBUFC) /MARK SEG CHANGED
/ENTRIES TO LOOP
TYIBL2,   LIO I (ATEM+3) /RESTORE G ETC
TYIBL4,   DIO I (ATEM+2) /PTR BEG THIS # FOR TYIS
          LAI
          IDC
          DAC G
TYIBL7,   LCH G
          SAS ML4        /NEEDED FOR ←L#-EOM WHERE # FAILS
          SZF I 1        /CLEARED WHEN EOM SEEN (OTHER CASES)
          JMP TYIREA     /GO RESTART IF PF1 CLEAR
          SZF 5
          JMP TYIBLQ     /IF LOOKING FOR A # AFTER A -
TYIBL8,   DZM I (ATEM+6) /NO # TO MARK START
          SZF I 3
          JMP TYIBLQ     / - LEGAL ONLY IF ←L
          SAD (CHARAC L-)
          JMP TYIBLE     /LIST "ELEMENT" STARTS WITH -
TYIBLQ,   CLL↑UCML       /LINK SET FOR TYIS
          JSP DQN        /DECODE QUESTION NUMBER
           JMP TYIBLI    /FORMAT ERROR
          LAC G
          DAC I (ATEM+3)
          LCH G
          SZF 3
          SZF 5
          JMP TYIBLS     /SECOND - NOT ALLOWED
          SAD (CHARAC L-)
          JMP TYIBLM     / # FOLLOWED BY -
TYIBLS,   SAD CHLCM
          JMP TYIBLM
          SAS ML4
           JMP TYIBLI    /FORMAT ERROR
          CLF 1
TYIBLM,   JSP SEARCH
           JMP TYIBLK     /NO ENTRY
          LAC E
          DAC D
          IDX D
          LAC I D        /GET ANSWER PTR
          SAD (-Ø)
           JMP TYIBLV     /NOT ANSWERED.
          SZF 3          /ON ←# ETC,
TYIBLB,   SZF 5          /OR IF THIS WAS THE # AFTER - ,
          JMP TYIBL9     /GO TO THE END OF THE LOOP, MARK THIS QUES STOP
↑L
```

```
          LIO I (ATEM+3) / # AT START OR AFTER , IN ↑L IF HERE
          DIO G
          LAC E
          DAC I (ATEM+7) /SAVE PTRS FOR MARKING ON ENTRY
          LAC I (JQBUFC) /  AFTER WHOLE LIST EL. VERIFIED
          DAC I (ATEM+6)
          LCH G
          SAD (CHARAC L-)
          JMP TYIBLJ
          JMP TYIBL9      /LOOP BACK. NOTE THAT E NOT ZEROED, SO
                          /THIS QUES WILL BE MARKED BOTH START AND STOP

TYIBLI,   LAW TYIBL6      /ILLEGAL FORMAT
          JMP TYIBIS      /TYPES # AND COMMENT

TYIBLK,   LAW TYIBL6      /QUES NOT ASKED
          JMP TYIBKS      /ERROR COMMENT

TYIBLV,   LAW TYIBL6      /NOT ANSWERED (EG CURRENT QUESTION)
          JMP TYIBAS
↑L
```

```
/←/
TYIBS,    LCH I G
          SAD ML4
          JMP TYIBSN
          SAD (CHARAC LS)
          LIO B17
          SAD CHARLL
          LIO B16
          SAD CHARLH
          LIO (3)          /INSERT TAG IF NOT THERE
          LCH I G
          SNI I            /IO Ø MEANS NO ACCEPTABLE CHAR FOUND
          SAS ML4          /INSIST ON EOM
          JMP TYIBR
TYIBSN,   LAC I (JBF)
          DIO I (JBF)
          CMI
          AAI
          SPA
          DZM I (JLQNT)    /RETYPE QUESTION IF JBF DOESN'T DECREASE

/←COMMANDS COME HERE TO RESTART
TYIREA,   LIO C67          /(67) FAST RESTART FLAGS
          SZF 2
          LIO C64          /(64). FULL RESTART IF A CHANGE HAS BEEN MADE
          SZF 4
          SZF I 3
          SKP I
TYIBLU,   LIO C65          /IN LIST MODE ON PF3&4 OR INITIAL EOM
          JMP KILLFI       /JMP TO KILL IOT



/←N RESTART FROM GQN IN NULLING MODE
TYIBN,    LCH I G
          SAS ML4          /EOM REQUIRED
          JMP TYIBR
          LIO (63          /RESTART IN NULL MODE
          JMP KILLFI
↑L
```

```
/←H
TYIBH,    LCH I G
          RCL 6S
          LCH I G
          RCL 6S
          LCH I G
          AAI             /PUTS 3 CHAR IN AC IN ORDER 3-1-2
          SAS (FLEXO #OW)          /CHECK FOR EXACTLY "←HOW#"
          JMP TYIBR
          JSP SPAC        /SEPERATING SPACE
          LAC I (JLGQN)   /SPO+LOC(LAST GQN)
          ADD (IOR)       /MAKES PTR TO LOC(LAST GQN)+1
          LIA
          JSP TRACEI
          IDX A
          IDX A           /GET HOW PTR
          LIO I A
          JSP JTOS
          JMP TYIREA

TYIBM,    LCH I G
          SAS ML4         /CHECK FOR EXACTLY "←-#"
          JMP TYIBR
TYIBM1,   LAW JHTSU       /USED BY ERROR ROUTINE TOO
          JMP GO

/←#
TYIVOK,   LAW 1
          SAS I (JMODE)   /TOC MODE?
          JMP TYINVF      /NO, ILLEGAL, TYPE FIX, REASK
          LAC I (JTBUFP) /TOC MODE, PRESERVE OLD ANSWER
          DAC I (JOJM)    /SO JTBUFP DOESN'T CHANGE
          LAC I E
          DAC D
          JSP .GTEXT
          LAC D
          JMP TYIBRB
↑L
```

```
/←C
TYIBC,    LCH I G
          SAD (CHARACTER LR)
          JMP TYIBCR
          RAL 6S
          XOR B13          /DIGIT?
          SUB C10.         /(10.)
          SMA
          JMP TYIBR        /NO, GIVE IT TO USER
          JSP DQN
           JMP TYIBCI
          LCH G
          SAS ML4
           JMP TYIBCI
          JSP SEARCH
           JMP TYIBCK
          IDX E
          LAC I E
          SAD (-0)
           JMP TYIBCA      /NOT ANSWERED
          DAC D            /MOBY PTR
          LAC I (ATEM+5)   /SEG
          DAC F
          JSP SPAC
          JSP READTQ       /LINK SET FROM SEARCH
          JSP .GTEXT       /GET OLD ANSWER
          LAC I (JTBUFP)
          DAC I (JOJM)
          LIO D            /CORE PTR TO OLD ANSWER
          JSP JTOS
          LAC D
          JMP TYIBRC       /RETURN TO TYINV IOT

TYIBCR,   LCH I G          /CHECK FOR ←CRASH
          RCL 6S
          LCH I G
          RCL 6S
          LCH I G
          AAI
          SAS (FLEXO HAS)          /ASH BUT OUT OF ORDER
          JMP TYIBR
          LCH I G
          SAS ML4          /IS IT FOLLOWED BY #
          JMP TYIBR
          765432           /EXECUTE ILLEGAL INSTRUCTION
↑L
```

```
/TYOC IOT
TYOC,     JSP TRACE
          LIO A
          SZF I 6
          JMP TYOCC
          LIO I (JBF)     /JBF DEPENDENT OPTION
          RIR 2S
          SPI
          IDX A
          LIO I A
TYOCC,    LAC I (JMODE)
          SMA
          JSP JTOS        /DON'T TYPE IN SIM OR NULL MODES
          JMP R1

B1,       LAC
C10.,     10.

REPEAT 0IF P,PRINT .START OF CONSTANTS.
CON,      CONSTANTS
FOO:
REPEAT 0IF P,PRINT .PATCH AREA.
REPEAT 1IF VP FOO-170001,PRINT .CORE OVERFLOW.
REPEAT 0IF VP FOO-170001,FLEXO FOO

START
↑L
```