ANIMAL Source Code

by John Walker

The following is the original UNIVAC® 1100 series assembly language for ANIMAL, the host program which used PERVADE to propagate itself among the UNIVAC installed base. The program is, itself, unremarkable. Note that the assembly-time variable which controlled whether or not it pervaded was named “VIRUS”.

In 1975 each computer vendor had their own terminology for concepts such as files, programs, etc. which have since become (reasonably) standardised. The following brief lexicon gives contemporary translations of Univac mainframe speak of the 1960's and 70's you'll encounter reading the code.

Univacky	Modern Term
element	file
ER	system call
PCT	open file table
processor	application
run	job or session

And of course, back the days when we were just making the transition from card punches to timesharing terminals, **real programmers** wrote in ALL CAPITALS. UNIVAC old-timers who have forgotten some of the instruction mnemonics may want to refer to the instruction set summary. This was developed on an 1110, but would run on the 1106 and 1108 as well. I could not find a copy of this in machine-readable form, so I had to type it in from a 20 year old listing. I've tried to proofread it carefully, but there may be some typos still lurking herein.

```
.
.         T H E   A N I M A L   G U E S S I N G   P R O G R A M
.
.                                       JOHN WALKER   APRIL 1974
.
.         (OR, BRUTE FORCE ARTIFICIAL INTELLIGENCE)
.
.         THIS PROGRAM GIVES THE APPEARANCE OF POSSESSING INTELLIGENCE
.         BY BEING ABLE TO GUESS ANIMALS THOUGHT OF BY THE USER.  IT
.         ASKS QUESTIONS AND FINALLY TELLS THE USER WHICH ANIMAL HE
.         HAD IN MIND.  IF IT IS INCORRECT, IT ASKS THE USER WHICH
.         ANIMAL HE HAD IN MIND AND ASKS HIM TO SUPPLY A QUESTION
.         WHICH DISTINGUISHES THE ANIMAL THE PROGRAM THOUGH WAS
.         CORRECT FROM THE USER'S ANIMAL.  THIS INFORMATION IS THEN
.         SAVED IN THE ANIMAL MEMORY FILE: HENCE THE PROGRAM LEARNS
.         THROUGH EXPERIENCE.
.
.         THE PROGRAM USES RANDOM SELECTION OF VARIOUS REPLIES AND
.         SUBSTITUTION OF NOUNS FOR PRONOUNS IN SENTENCES TO AVOID
.         THE REPETITIOUS PATTER OUTPUT BY MOST INTERACTIVE QUERY
.         PROGRAMS.  THE PROGRAM IS CAPABLE OF DETECTING IF IS
.         BEING LED ASTRAY IN THE DESCRIPTION OF AN ANIMAL, AND IF
.         SO, ASKS THE USER TO MAKE SURE HIS DESCRIPTION JIBES WITH
.         THE DESCRIPTION THE PROGRAM ALREADY HAS.  WHAT THE USER
.         TELLS THE PROGRAM THREE TIMES IS TRUE, AND THE PROGRAM
.         WILL UNLEARN AN INCORRECT DESCRIPTION OF AN ANIMAL.
.
.         THE ANIMAL GUESSING PROGRAM IS SELF-INSTALLING AND SELF-
.         MAINTAINING.  IT CREATES AND UPDATES A MEMORY FILE AS IT
.         IS USED.  THE NAME OF THIS MEMORY FILE MAY BE CHANGED
.         BY THE GENERATOR OF THIS PROGRAM BY MODIFYING THE FILE
.         NAMES FOUND ON THE 'MEMFILE' AND 'BACKUPFN' FOLLOWING
.         THIS TEXT AND REASSEMBLING THE PROGRAM.
.
.         IF GENERATED WITH THE TAG 'MAINTENANCE' (SEE BELOW) SET
.         NONZERO, IT IS POSSIBLE TO SIGN ON TO ANIMAL WITH THE
.         'M' OPTION AND THE KEYS TO THE ANIMAL MEMORY FILE SPEC-
.         IFIED ON THE CALL STATEMENT:
.
.                   @ANIMAL,M RKEY/WKEY
.
.         AND ENTER 'ANIMAL FILE MAINTENANCE MODE'.  IN THIS MODE,
.         THE USER MAY TYPE IN COMMANDS AND 'EDIT' THE ANIMAL TREE.
.         THE OPERATIONS AVAILABLE ARE:
.
.                   AB                  ABORT MAINTENANCE, DON'T UPDATE FILE
.                   CA                  CHANGE NAME OF ANIMAL
.                   CQ                  CHANGE QUESTION TEXT
.                   DA                  DELETE ANIMAL FROM MEMORY
.                   DQ                  DELETE QUESTION FROM MEMORY
.                   GA                  ADD GENERIC NAME TO PROHIBITED LIST
.                   GD                  DELETE GENERIC NAME FROM PROHIBITED LIST
.                   LT                  LIST MEMORY TREE
.                   PL                  PLAY A ROUND OF THE GAME
.
.         WHEN USING ANY OF THE COMMAND WHICH OPERATE ON A SPECIFIC
.         NODE, SUCH AS CA, DA, OR CQ, THE USER WILL BE PUT INTO GAME
.         MODE SO THAT HE CAN LEAD THE PROGRAM TO THE ANIMAL OR QUESTION
.         TO BE DELETED OR CHANGED.  WHEN THE DESIRED QUESTION OR ANIMAL
.         IS OUTPUT BY THE PROGRAM, THE USER SHOULD TYPE:
.
.                   THAT'S IT
.
.         TO THE PROGRAM RATHER THAN THE NORMAL YES OR NO ANSWER.  THAT
.         WILL SELECT THE NODE FOR PROCESSING.  WHEN IN MAINTENANCE MODE,
.         ALL QUESTIONS AND ANIMALS PRINTED WILL BE PRECEDED BY THEIR
.         RELATIVE MEMORY ADDRESS, WHICH CAN BE USED TO FIND THEM IN
.         THE TREE LISTING PRODUCED BY THE 'LT' COMMAND.
.
.                   END                 END MAINTENANCE, UPDATE FILE
.
MAINTENANCE EQU     1                   ENABLE MAINTENANCE MODE
.
.
.         THE TAG 'LEVEL' DEFINED THE LEVEL OF THE ANIMAL PROCESSOR.
.         THIS NUMBER IS DISPLAYED WHEN THE USER SIGNS ON IN MAINTENANCE
.         MODE, AND IS KEPT IN THE FILE FOR PURPOSES OF COMPATIBILITY
.         BETWEEN LEVELS (IN OTHER WORDS, THE AUTOMATIC CONVERSION
.         OF OLD ANIMAL FILES TO THE NEW FORMAT WHEN A NEW ANIMAL
.         PROCESSOR LEVEL IS IMPLEMENTED).  THIS TAG SHOULD NOT BE
.         CHANGED IN THE FIELD.
.
LEVEL     EQU       '2.0'               LEVEL OF ANIMAL PROCESSOR
.
.
```

```
.         THE TAG 'LOCLVL' DEFINES A LOCAL LEVEL OF THE ANIMAL PROCESSOR.
.         IF ANIMAL IS REGENERATED BY A SITE, ANT THE SITE WANTS TO
.         IDENTIFY THEIR LOCAL LEVEL, THEY SHOULD USE THIS TAG, NOT
.         THE TAG 'LEVEL' ABOVE.  THIS VALUE WILL BE KEPT IN THE ANIMAL
.         MEMORY FILE IMMEDIATELY AFTER THE VALUE FOR 'LEVEL' AND WILL
.         BE CONCATENATED WITH 'LEVEL' IN THE THE MAINTENANCE MODE SIGN-ON
.         LINE.  THIS VALUE WILL ALWAYS BE ZERO IN SYMBOLICS SHIPPED
.         BY ANIMAL DEVELOPMENT CENTRE (ADC).
.
LOCLVL    EQU       0                   NO LOCAL CODE IN ADC SOFTWARE !
.
.         THE TAG 'VIRUS' CONTROLS WHETHER THE ANIMAL PROCESSOR WILL SPREAD
.         LIKE ONE THROUGHOUT THE 1100 SERIES USER COMMUNITY.
.
VIRUS     EQU       1                   GO WILD !
.
          LIT$      2
.
.         THE FOLLOWING LINE DEFINES THE NAME OF THE CATALOGUED ANIMAL
.         MEMORY FILE.  THIS MAY ME CHANGES AT THE WHIM OF THE GENERATOR
.         OF THE ANIMAL PROCESSOR.  THE FORMAT FOR THE FILE NAME MUST
.         BE: 'QUAL*FILENAME&/RKEY/WKEY/&'
.
MEMFILE   'BOBO*SIMPLEMINDED&/DUDLEY/DORITE&'
.
.         THE FOLLOWING LINE DEFINES THE NAME OF THE BACKUP FILE FOR THE
.         ANIMAL MEMORY.  THE BACKUP FILE IS USED TO ATTEMPT RECOVERY
.         IF THE ANIMAL MEMORY FILE IS DESTROYED.  IF THE TAG
.         BACKUPFN IS EQUATED TO ZERO, THE BACKUP MECHANISM WILL BE
.         TURNED OFF AND THE CODE REMOVED FROM THE ANIMAL PROCESSOR.
.
BACKUPFN  '853429*EIGENVALUE&/VECTOR/CALCUL&' BE INCONSPICUOUS
.
.
.
          AXR$
          DEFUNCT$
          ELT$
.
EOL       EQU       077                 LINE TERMINATOR CHARACTER
.
          CHAR      '%',EOL             DEFINE STOP CHARACTER
.
.         STRUCTURE OF NODE IN TREE
.
NODELNK   EQUF      0                   YES AND NO LINK WORD
NODEYL    EQUF      NODELNK,,H1         YES LINK
NODENL    EQUF      NODELNK,,H2         NO LINK
NODELEN   EQUF      1,,S1               LENGTH OF NODE TEXT
NODEREFC  EQUF      1,,H2               NODE REFERENCE COUNT (FOR REBALANCE)
NODEBL    EQUF      2,,H1               BACK LINK TO FATHER NODE
NODEBITS  EQUF      2,,H2               CONTROL BITS FOR THIS NODE
NODEBKL   EQUF      3,,H1               BACK LINK TO PREVIOUS NODE
NODEFL    EQUF      3,,H2               FORWARD LINK TO NEXT NODE
NODEUID   EQUF      4                   USERID OF NODE ADDER
NODEAC    EQUF      5                   ACCOUNT NUMBER OF NODE ADDER
NODETEXT  EQUF      7                   START OF TEXT IN NODE
.
.         MEANING OF BITS IN 'NODEBITS'
.
NBDEL     EQU       1                   NODE IS DELETED
NBGENERIC EQU       2                   NODE IS A PROHIBITED GENERIC NAME
```

```
.
.         FUNCTION FOR REFERENCING MEMORY WITH RELATIVE ADDRESSES
.
F         FUNC      .
M*        NAME      0
          END       MEMORY+F(1)
$(1).
.
BEGIN     SR        R2,R15              SAVE TDATE$ AT PROCESSOR CALL
          ON        VIRUS
          LMJ       X11,PERVADE         PERVADE THROUGHOUT THE FILE SYSTEM
          OFF       VIRUS
          LA        A0,(INFL,INFOR)     LOAD INFOR BUFFER ADDRESS
          LMJ       X11,RINF$           READ INFOR TABLE
          PRINT$    .                   PRINT ERROR MESSAGE IF ANY
.
.         ACQUIRE THE MEMORY FILE
.
          ON        MAINTENANCE
          LA        A1,INFOR            LOAD OPTIONS FROM CALL STATEMENT
          LA,U      A0,1                LOAD INFOR FIELD NUMBER
          TEP,U     A1,OPTION('M')      IS THE 'M' OPTION ON ?
          LMJ       X11,SELT$           LOOK FOR FIELD 1 IN INFOR
          J         NOMAINT             SKIP IF NO FIELD SPECIFIED
          TZ        ENL                 ELEMENT NAME PRESENT ?
          TNZ       EVL                 ELEMENT VERSION PRESENT ?
          J         NOMAINT             NO.  IGNORE THE 'M' OPTION
          LA,U      A0,1                LOAD A ONE
          SA        A0,MAINT            SET MAINTENANCE MODE FOR PROCESSOR
NOMAINT   OFF       MAINTENANCE
ASGMA     F$MSG1    ASGMEM              EDIT THE MEMORY FILE ASSIGN IMAGE
          F$MSG     MEMFILE             EDIT THE MEMORY FILE NAME
          ON        MAINTENANCE
          TNZ       MAINT               IN MAINTENANCE MODE ?
          J         ASGUSR              NO.  USER STANDARD KEYS ON FILE
          F$CHAR    '/'                 EDIT A SLASH BEFORE THE READ KEY
          F$FD1     ENAME               EDIT ELEMENT NAME AS READ KEY
          F$CHAR    '/'                 EDIT A SLASH BETWEEN KEYS
          F$FD1     EVER                EDIT ELEMENT VERSION AS WRITE KEY
          J         ASGMF               GO ASSIGN THE MEMORY FILE
ASGUSR    OFF       MAINTENANCE
          F$MSGR    .                   COPY THE READ AND WRITE KEYS
ASGMF     CSF$      FL$                 TRY TO ASSIGN THE FILE
          JP        A0,MEMOK            SKIP IF ASSIGNED CORRECTLY
          TEP       A0,(BIT(21))        DOES ANIMAL NEED TO BE INSTALLED ?
          J         INSTALL             YES.  INSTALL ANIMAL AT THIS SITE
          TOP       A0,(BIT(18))        FACILITY WAIT STATUS ?
          J         LATER               NO.  TELL USER TO TRY LATER
          TWAIT$    10000               WAIT FOR TEN SECONDS
          CSF$      FL$                 TRY ONE MORE TIME
          JP        A0,MEMOK            SKIP IF OK THIS TIME
LATER     PRINT$    LATEM,LATEL
          J         ENDALL              TERMINATE
.
.         ENTER THIS CODE FOR UNEXPECTED EOF
.         UPDATES WILL NOT BE APPLIED TO THE FILE
.
EOFANS    CSF$      FREEMEM             FREE THE MEMORY FILE
          ON        MAINTENANCE
          TNZ       MAINT               IN TREE MAINTENANCE MODE ?
          J         EOFANM              NO.  EDIT NORMAL MESSAGE
          PRINT$    ENDMM,ENDML         PRINT TREE MAINTENANCE END MESSAGE
          J         ENDALL              TERMINATE MAINTENANCE MODE
EOFANM    OFF       MAINTENANCE
          PRINT$    EOFAM,EOFAL         PRINT ALTERNATE SIGN OFF MESSAGE
ENDALL    .
          ON        VIRUS
          LMJ       X11,PVTERM          TERMINATE PERVASION IF IN PROGRESS
          OFF       VIRUS
          EXIT$     .                   THAT'S ALL
.
.         PRINT THE SIGN-ON MESSAGE  (THIS OVERLAPS WITH READ OF MEMORY FILE)
.
MEMOK     .
          ON        MAINTENANCE
          TZ        MAINT               MAINTENANCE MODE ?
          J         MASKSO              YES.  SKIP NORMAL SIGN-ON
          OFF       MAINTENANCE
          PRINT$    SIGNON,SIGNL        PRINT THE SIGN-ON-LINE
.
.         READ THE MEMORY FILE AND BUILD THE IN-CORE TREE
.
MASKSO    F$DT      .                   CLEAR THE LINE
          F$MSG1    USEMEM              EDIT THE @USE IMAGE
          F$MSG1    MEMFILE             INSERT THE MEMORY FILE NAME
          CSF$      FL$                 ATTACH THE @USE NAME TO THE MEMORY FILE
          IOW$      IOP                 READ SECTOR ZERO OF MEMORY FILE
          F$DT      .                   CLEAR THE EDITING IMAGE
          TZ,S1     IOP+3               NORMAL COMPLETION ?
          J         LATER               I/O ERROR READING MEMORY
          LA        A0,MEMORY           LOAD FILE SENTINEL
          TE        A0,('ANIMAL')       IS IT A VALID ANIMAL FORMAT FILE ?
          J         LATER               NO.  REINITIALISE
          LA        A1,MEMLEN           LOAD MEMORY LENGTH IN WORDS
          LA,U      A0,MEMORY+200,A1    COMPUTE HIGHEST ADDRESS NEEDED
          SA        A0,HIGHCORE         SET LARGEST ALLOCATED ADDRESS
          MCORE$    .                   EXPAND PROGRAM TO ACCOMMODATE FILE
          SA,H1     A1,IOP+4            SET READ LENGTH FOR ACTUAL READ
          IOW$      IOP                 READ THE MEMORY FILE INTO CORE
          TZ,S1     IOP+3               NORMAL STATUS ?
          J         LATER               NO.  REINITIALISE THE FILE
          LA        A0,MEMLEN           LOAD HIGHEST ADDRESS ASSIGNED IN MEMORY
          AA,U      A0,MEMORY           ADD BASE ADDRESS OF MEMORY
          SA        A0,HIGHUSE          SAVE HIGHEST ADDRESS IN USE
          PCT$,0    USERID              GET THE RUNID (USERID)
          PCT$,023  ACCOUNT,2           SAVE THE ACCOUNT NUMBER
          ON        MAINTENANCE
          TZ        MAINT               IN MAINTENANCE MODE ?
          J         MAINTMAIN           YES.  ENTER MAINTENANCE COMMAND SCANNER
          OFF       MAINTENANCE
/.
.
.         NOW THE FUN BEGINS:  ASK QUESTIONS OF THE USER
.
          PRINT$    SIGNON1,SIGNL1      TELL THE USER WHAT'S COMING UP
RESTART   LX        X8,BASENODE         LOAD RELATIVE BASE NODE ADDRESS
.
.         EXECUTE THE NODE POINTED TO BY X8
.
DONODE    LX,U      X9,MEMORY,X8        COMPUTE ABSOLUTE ADDRESS OF CURRENT NODE
          TNZ       NODELINK,X9         IS THIS A QUESTION NODE ?
          J         ITHINK              NO.  THIS IS A LEAF NODE.  TELL
.                                       THE USER WHAT WE THINK IT IS.
.
```

```
.         THE NODE CONTAINS A QUESTION.  POSE IT TO THE USER
.
ASKIT     .
          ON        MAINTENANCE
          TZ        MAINT               IN MAINTENANCE MODE ?
          LMJ       X11,RELADR          YES.  EDIT ADDRESS OF NODE
          OFF       MAINTENANCE
          LA,U      A0,NODETEXT,X9      LOAD ADDRESS OF TEXT
          LA        A1,NODELEN,X9       LOAD LENGTH OF QUESTION
          F$COPY    .                   COPY QUESTION
          LMJ       X4,DECIDE           HAVE THE USER DECIDE
          J         ASKAY               USER SAID 'YES'
          LX        X8,NODENL,X9        LOAD NO LINK BECAUSE USER SAID 'NO'
          J         DONODE              GO PROCESS NEXT NODE
.
ASKAY     LX        X8,NODEYL,X9        LOAD THE 'YES' LINK
          J         DONODE              PROCESS THE NODE
/.
.
.         THIS NODE HAS NO LINKS.  TELL THE USER WHAT WE THINK IT IS
.
ITHINK    F$MSG     ISIT                EDIT 'IS IT'
          ON        MAINTENANCE
          TZ        MAINT               IN MAINTENANCE MODE ?
          LMJ       X4,RELADR           YES.  EDIT ADDRESS OF NODE
          OFF       MAINTENANCE
          LA        A1,NODELEN,X9       LOAD LENGTH OF NODE ENTRY
          LA,U      A0,NODETEXT,X9      LOAD TEXT START ADDRESS
          F$COPY    .                   COPY TEXT TO THE BUFFER
          LMJ       X4,DECIDE           ASK THE USER YES OR NO
          J         AGAING              ASK USER IF HE'S TIRED YET
.
.         WE GUESSED AND THE USER CLAIMS THAT OUR ANIMAL WAS WRONG.
.         ASK HIM WHAT ANIMAL HE HAD IN MIND.
.
          LX,U      X6,WHATANI          LOAD QUESTION CONTROL PACKET ADDRESS
          LMJ       X5,QUESTION         ASK HIM WHAT ANIMAL IT WAS
.
.         NOW SAVE HIS ANIMAL AND ASK FOR A QUESTION TO
.         DISTINGUISH HIS ANIMAL FROM THE ONE WE GUESSED.
.
.
.         SEE IF AN ARTICLE WAS SUPPLIED BY THE USER.  IF NOT,
.         GENERATE ONE.
.
ITHANS    LMJ       X5,SCANANI          SCAN THE ANIMAL SUPPLIED BY THE USER
.
.         AT THIS POINT WE HAVE ACCEPTED THE USER'S ANIMAL AND REDUCED IT
.         TO CANONICAL FORM.  WE SCAN THE LINEAR LIST OF NODES AND SEE IF
.         THE ANIMAL THE USER TYPED IN DUPLICATES ANY ANIMAL WE ALREADY
.         HAVE IN THE TREE.  IF SO, WE ENTER SPECIAL PROCESSING FOR
.         DUPLICATE ANIMALS BELOW.
.
          LX,H2     X5,NODECHAIN        LOAD HEAD OF LINEAR NODE CHAIN
.
ALRSCN    TZ        NODEYL+MEMORY,X5    IS THIS A LEAF NODE ?
          J         ALRQUN              NO.  IGNORE IT
          TE        A15,NODELEN+MEMORY,X5 ARE LENGTHS THE SAME ?
          J         ALRQUN              NO.  THEY CANNOT BE EQUAL
          LR        R1,UANLW            LOAD USER'S ANIMAL LENGTH IN WORDS
          LA        A0,(1,0)            LOAD POINTER TO SCAN ANIMAL
          LA        A1,(1,0)            LOAD POINTER TO USER'S ANIMAL
          AA,U      A0,NODETEXT,X5      FORM POINTER TO TEXT OF SYMBOL
          J         ALRCMPE             ENTER COMPARISON LOOP
.
ALRCMPS   LA        A2,MEMORY,*A0       LOAD WORD FROM MEMORY ANIMAL
          TE        A2,UANML,*A1        COMPARE WITH USER'S ANIMAL
          J         ALRQUN              UNEQUAL.  IT'S NOT THIS ANIMAL
ALRCMPE   JGD       R1,ALRCMPS          LOOP FOR ALL WORDS IN ANIMAL
          LA        A0,M(NODEBITS),X5   LOAD TYPE BITS FROM FIND NODE
          TEP,U     A0,NBDEL            IS THE NODE WE FOUND THE USER'S
.                                       ANIMAL IN ACTUALLY A DELETED NODE ?
          J         ALRQUN              YES.  IGNORE THE FIND
          TEP,U     A0,NBGENERIC        WAS USER'S ANIMAL FOUND AS A
.                                       PROHIBITED GENERIC TYPE ?
          J         GENERIC             YES.  ASK HIM TO BE MORE SPECIFIC
          J         ALRDUP              FIND.  HANDLE DUPLICATE ANIMAL
.
.         LINK TO NEXT NODE IN THE TREE
.
ALRQUN    LA        A0,X5               LOAD NODE RELATIVE ADDRESS
          TNE       A0,MEMORY+NODEBKL   WAS THIS THE LAST NODE IN THE TREE ?
          J         GMQA                YES.  USER'S ANIMAL IS NEW TO US
          LX        X5,NODEFL+MEMORY,X5 LINK TO NEXT NODE IN TREE
          J         ALRSCN              SCAN IT FOR EQUALITY
.
.         USER'S ANIMAL IS A GENERIC TYPE WE'VE BEEN WARNED ABOUT.  FORCE
.         HIM TO BE MORE SPECIFIC ABOUT WHAT KIND OF ANIMAL HE HAS IN
.         MIND.
.
GENERIC   LX,U      X6,SPECIFY          LOAD 'PLEASE BE MORE SPECIFIC' QUESTIONS
          LMJ       A2,RANFDIT          EDIT A QUERY FROM OUR SET
          LA,U      A0,UANML            LOAD USER'S ANIMAL
          LA        A1,UANL             LOAD LENGTH OF USER'S ANIMAL
          F$COPY    .                   APPEND USER'S ANIMAL TO THE QUESTION
          LMJ       X5,QUESTK           ASK USER WHICH SPECIFIC ANIMAL
          J         ITHANS              INTERPRET HIS NEW ANSWER
.
.
.         NOW THAT WE HAVE ESTABLISHED THAT THE USER'S ANIMAL IS UNIQUE
.         IN THE TREE, OR HAVE REMOVED ANOTHER INSTANCE OF HIS ANIMAL
.         WHICH HE CLAIMS WAS WRONGLY DESCRIBED, WE ENTER THIS SECTION
.         OF CODE WHICH ASKS THE USER FOR A QUESTION WHICH WILL
.         DISTINGUISH HIS ANIMAL FROM THE ONE WE THOUGHT WAS CORRECT.
.
GMQA      F$MSG     GMQ                 EDIT TEXT FOR QUESTION
          LA,U      A0,UANML            LOAD ADDRESS OF REPLY
          LA        A1,UANL             LOAD LENGTH OF USER'S ANIMAL
          F$COPY    .                   COPY ANIMAL INTO QUESTION
          F$MSGR    .                   COPY SOME MORE QUESTION
          LA,U      A0,NODETEXT,X9      LOAD ADDRESS OF GUESSED ANIMAL
          LA        A1,NODELEN,X9       LOAD LENGTH OF THAT ANIMAL
          F$COPY    .                   COPY GUESSED ANIMAL TO MESSAGE
          F$CHAR    ':'                 EDIT COLON
          F$PRT     1                   PRINT THE QUESTION
ASKQAG    READ$     REPLY,EOFANS        READ THE USER'S ANSWER
          TNZ,U     0,A0                VOID ANSWER ?
          J         GMQA                YES.  ASK AGAIN
          LMJ       X5,QUESTL           SCAN THE PURPORTED QUESTION
.
.         SCAN THE SUBMITTED QUESTION
.
          LMJ       X4,SCANQUES         SCAN THE QUESTION FROM THE USER
          J         ASKQAG              REASK IF SUBMITTED QUESTION INCORRECT
          LMJ       X7,PLUGGEN          LOOK FOR PRONOUN IN QUESTION
          J         REASTFD             NONE.  ASK GENERAL QUESTION
.
.         NOW THAT WE HAVE PLUGGED THE USER'S QUESTION TO THAT AN ANIMAL
.         MAY BE INSERTED INTO THE QUESTION IN PLACE OF THE PRONOUN,
.         WE RANDOMLY DECIDE WHETHER TO INSERT THE USER'S ANIMAL OR THE
.         ANIMAL FROM THE TREE INTO THE QUESTION.  THIS NOT ONLY MAKES
.         THE PROGRAM APPEAR MORE INTELLIGENT, BUT OFTEN LEADS TO
.         UNEXPECTED HUMOUR, E.G.,
.
.         CAN <A PYTHON> COMPETE IN THE KENTUCKY DERBY?
.
.         WHERE THE USER'S ANIMAL WAS A HORSE.
.
          TIME$     .                   GET TIME OF DAY
          AND,U     A0,1                ISOLATE LOW ORDER BIT
          SA        A1,A14              SAVE INVERSION FLAG FOR REPLY
          F$MSG     REPLY               EDIT THE USER'S QUESTION BACK AT HIM
          JB        A14,GMQWOA          USE USER'S ANIMAL IN THE QUESTION ?
          LA,U      A0,UANML            YES.  LOAD USER ANIMAL ADDRESS
          LA        A1,UANL             LOAD USER ANIMAL LENGTH
          J         GMQIAN              GO COPY THE REST OF THE QUESTION
.
GMQWOA    LA,U      A0,M(NODETEXT),X8   LOAD ADDRESS OF OUR ANIMAL
          LA        A1,M(NODELEN),X8    LOAD LENGTH OF OUR ANIMAL
.
GMQIAN    F$COPY    .                   COPY ANIMAL INTO THE QUESTION
          F$MSGR    .                   IGNORE THE SECOND FMSG$ STOP
          F$MSGR    .                   COPY THE REST OF THE QUESTION
REASKD    LA,U      A13,1               INITIALISE A13 FOR 'NO' REPLY
          LMJ       X4,DECIDE           WOULD USER ASK HIS QUESTION 'YES'
.                                       FOR HIS ANIMAL ?
          LA,U      A13                 YES.  FLAG TO SET LINKS FROM QUESTION
          XOR       A14,A13             INVERT USER'S ANSWER IF WE ASKED
.                                       THE QUESTION WITH OUR ANIMAL.
.
.         INCLUDE THE USER'S ANIMAL IN THE MEMORY TREE
.
          LA        A0,UQL              LOAD LENGTH OF USER'S QUESTION
          LMJ       X11,MAKENODE        BUILD NODE FOR USER QUESTION
.
.         CONSTRUCT LINKS IN USER QUESTION NODE
.
          LX        X1,MEMORY+NODEBL,X8 LOAD BACK LINK FROM OUR ANIMAL
          LA,U      A4,,X8              LOAD ADDRESS OF ANIMAL WE THOUGHT IT WAS
          TNE       A4,MEMORY+NODEYL,X1 WAS ANIMAL OFF 'YES' LINK ?
          SA        A1,MEMORY+NODEYL,X1 YES.  ATTACH NEW QUESTION TO 'YES' LINK
          TNE       A4,MEMORY+NODENL,X1 WAS ANIMAL OFF 'NO' LINK ?
          SA        A1,MEMORY+NODENL,X1 YES.  ATTACH USER QUESTION TO 'NO' LINK
          SA        A1,NODEBL+MEMORY,X8 ATTACH PREVIOUS ANIMAL TO THIS QUESTION
.
.         PREVIOUS NODE NOW POINTS TO THE QUESTION SUPPLIED BY THE
.         USER WHICH DISTINGUISHES HIS ANIMAL FROM THE ONE WE HAD
.         IN THE TREE.
.
          LR        R1,UQLW             LOAD USER QUESTION IN WORDS
          LA        A3,(1,0)            LOAD DESTINATION POINTER
          AA,U      A3,NODETEXT+MEMORY,A1 GET TEXT ADDRESS
          LX        X11,(1,UQUES)       LOAD SOURCE POINTER
          BT        A3,,*X11            COPY USER QUESTION TO NEW NODE
          SX        X1,M(NODEBL),A1     SET QUESTION BACK LINK IN USER'S QUESTION
.
```

```
.         CREATE A LEAF NODE FOR THE USER'S NEW ANIMAL
.
          SA        A0,X1               SET NEW QUESTION AS PREVIOUS NODE
          LA        A0,UANL             LOAD USER'S ANIMAL LENGTH
          LMJ       X11,MAKENODE        CREATE A NODE FOR USER'S ANIMAL
          SX        X1,M(NODEBL),A1     SET QUESTION LINK IN USER'S ANIMAL
          LA        A3,(1,0)            LOAD DESTINATION POINTER
          AA,U      A3,NODETEXT+MEMORY,A1 POINT TO TEXT OF LEAF NODE
          LX        X11,(1,UANML)       LOAD SOURCE POINTER
          LR        R1,UANLW            LOAD USER'S ANIMAL LENGTH IN WORDS
          BT        A3,,*X11            COPY USER ANIMAL TO LEAF NODE
          LA,U      A0,,X8              NO LINK = OLD ANIMAL
          LXI,U     A0,,A1              YES LINK = NEW ANIMAL
          TZ        A15                 WAS ANSWER TO USER'S QUESTION 'YES'
.                                       FOR THE USER'S ANIMAL ?
          SSC       A0,18               NO.  INVERT LINKS IN QUESTION NODE
          SA        A0,NODELINK+MEMORY,X1 SET LINKS IN USER'S QUESTION
          J         AGAING              ASK THE USER IF HE'S DONE
.
.         WE COULD NOT FIND A PRONOUN IN THE USER'S QUESTION.
.         WE ARE FORCED TO ASK CIRCUITOUSLY WHAT THE CORRECT
.         ANSWER IS.
.
REASTPD   F$MSG     HOWDYA              ASK GENERAL QUESTION OF USER
          LA,U      A0,UANML            LOAD ADDRESS OF USER ANIMAL
          LA        A1,UANL             LOAD LENGTH OF USER ANIMAL
          F$COPY    .                   EDIT USER ANIMAL
          LA,U      A14                 INDICATE USER'S ANIMAL USED IN QUESTION
          J         REASKD              GO POSE THE QUESTION
/.
.
.         INPUT RECEIVING SUBROUTINES
.
.         THESE ROUTINES PERFORM THE PRELIMINARY CHECKING AND FORMATTING
.         OF ANIMALS AND QUESTIONS RECEIVED BY THE USER.  ALL SYNTAX AND
.         SEMANTIC CHECKING IS DONE BY THESE ROUTINES.
.
.
.         RECEIVE ANIMAL
.
.         ASSUMES THE USER REPLY IS IN THE BUFFER 'REPLY' AND A15 IS SET
.         TO THE REPLY LENGTH IN CHARACTERS (AS RETURNED BY THE 'QUESTION'
.         SUBROUTINE).
.
.         LMJ       X5,SCANANI
.         <RETURN>                      ANIMAL COPIED TO 'UANML'
.                                       UANL = ANIMAL LENGTH IN CHARACTERS
.                                       UANLW = ANIMAL LENGTH IN WORDS
.
.         ARTICLE PREFIXED TO ANIMAL IF USER DIDN'T SUPPLY ONE.
.
SCANANI   LA,U      A0,REPLY            LOAD FIRST WORD OF REPLY
          LA,U      A1                  CLEAR RESULT REGISTER
          LR,U      R1,5                LOAD LOOP COUNT FOR SCANNER
ITHFW     LDSC      A0,6                SHIFT OFF A CHARACTER
          AND,U     A0,077              ISOLATE LAST CHARACTER
          TE,U      A0,' '              IS IT A SPACE ?
          JGD       R1,ITHFW            LOOP FOR AT MOST 6 CHARACTERS
          SSL       A1,6                GET RID OF THE SPACE
          LR,U      R1,ARTTL            LOAD LENGTH OF ARTICLE TABLE
          LA        A0,(1,0)            LOAD POINTER TO TABLE
          SE        A1,ARTTT,*A0        IS FIRST WORD AN ARTICLE IN
```

```
.                                       ANY KNOWN HUMAN LANGUAGE ?
          J         $+2                 NO.  CHECK DOLPHIN AND CHIMPANZEE
          J         GMART               YES.  USER SUPPLIED AN ARTICLE
.
.         SUPPLY ARTICLE IF USER LEFT IT OUT
.
          F$DT      UANML,14            SET UP EDITOR ON REPLY BUFFER
          LA,S1     A2,REPLY            LOAD FIRST CHARACTER OF USER RESPONSE
          LR,U      R1,VOWELL           LOAD COUNT OF VOWELS
          LA        A1,(1,0)            LOAD SEARCH POINTER
          LA,U      A0,'AN'             LOAD ARTICLE FOR LEADING VOWEL
          SE        A2,VOWELS,*A2       IS FIRST CHARACTER A VOWEL ?
          LA,U      A0,'A'              NO.  USE 'A' AS ARTICLE
          F$FD1     .                   EDIT THE ARTICLE
          F$SKIP    1                   SKIP BEFORE THE REPLY
          LA,U      A1,A15              LOAD LENGTH OF USER'S ANIMAL
          LA,U      A0,REPLY            LOAD ADDRESS OF USER'S ANIMAL
          F$COPY    .                   COPY TO BUFFER AFTER ARTICLE
          F$COLN    .                   GET CHARACTERS EDITED
          SA        A0,A15              SET LENGTH OF USER'S ANIMAL
          DSL       A0,36               RIGHT JUSTIFY IN A0, A1
          AA,U      A1,5                ROUND FOR COVERED DIVIDE
          DI,U      A0,6                COMPUTE WORDS IN REPLY
          SA        A0,UANLW            SET LENGTH IN WORDS
          F$DT      FL$,FLL$            RESET EDITOR ON CANNED LINE
          J         GMARTS              ENTER COMMON CODE
.
GMART     LA        A0,A15              LOAD LENGTH OF REPLY
          DSL       A0,36               SHIFT FOR DIVIDE
          AA,U      A1,5                ROUND UP IN DIVIDE
          DI,U      A0,6                COMPUTE WORDS IN ENTRY
          SA        A0,UANLW            SAVE LENGTH IN WORDS
          LR        R1,A0               LOAD LENGTH IN WORDS
          LA        A0,(1,UANML)        LOAD SAVE BUFFER ADDRESS
          LA        A1,(1,REPLY)        LOAD POINTER TO REPLY
          BT        A0,,*A1             COPY REPLY TO SAVE BUFFER
GMARTS    SA        A15,UANL            SAVE LENGTH IN CHARACTERS
          J         0,X5                RETURN TO CALLER
.
.
.         SCAN QUESTION FROM USER
.
.         THIS SUBROUTINE PERFORMS ALL CHECKS FOR CORRECT FORMAT
.         OF QUESTIONS AND COPIES THE QUESTION TO THE BUFFER 'UQUES'.
.         THE CELLS 'UQL' AND 'UQLW' ARE SET TO THE LENGTH OF THE
.         QUESTION IN CHARACTERS AND WORDS RESPECTIVELY.
.
. CALL:   LMJ       X4,SCANQUES
.         <RETURN>                      QUESTION SYNTAX INVALID
.         <RETURN>                      QUESTION SCANNED PROPERLY
.
.
.
.         LOOK AT THE BEGINNING OF THE QUESTION AND SEE IF IT IS ONE
.         OF THE MOST COMMONLY USED BEGINNINGS OF QUESTIONS WHICH
.         CANNOT BE ANSWERED YES OR NO.  SUCH QUESTIONS, LIKE 'WHAT
.         HAS FOUR FEET' ARE FREQUENTLY TYPED IN BY PEOPLE WHO DON'T
.         COMPREHEND WHAT 'A QUESTION THAT DISTINGUISHES X FROM Y'
.         MEANS IN TERMS OF THE ENGLISH LANGUAGE.
.
SCANQUES  LA        A0,REPLY            LOAD FIRST WORD OF REPLY
          LR,U      R1,PREFXL           LOAD LENGTH OF PREFIX TABLE
          LR        R2,(0777777770000)  LOOK AT FIRST FOUR CHARACTERS
          LA        A1,(1,0)            LOAD SEARCH POINTER
          MSE       A0,PREFIX,*A1       LOOK FOR PREFIX IN TABLE
          J         $+2                 NOT AN ILLEGAL PREFIX.  QUESTION IS OK
          J         BADPREF             CLARIFY FOR USER IF BAD PREFIX
.
          E$DITR    REPLPK              SET UP THE SCANNER ON THE REPLY
          E$COL     0                   TAB TO COLUMN 1
GMQRMV    LA        A0,A15              LOAD CHARACTERS IN REPLY
          ANA,U     A0,1                CONVERT TO EDIT$ COLUMN
          E$COL     .                   TAB TO LAST CHARACTER IN QUESTION
          U$CHAR    .                   LOAD THE FINAL CHARACTER
          TE,U      A0,'?'              IS IT A QUESTION MARK ?
          TNE,U     A0,' '              NO.  IS IT A TRAILING SPACE ?
          J         $+2                 YES.  REMOVE IT
          J         GMQSL               NO.  VALID LAST CHARACTER, SET LENGTH
          ANA,U     A15,1               BACK UP LENGTH OF RESPONSE
          J         GMQRMV              LOOP FOR NEXT CHARACTER
GMQSL     SA        A15,UQL             SAVE USER QUESTION LENGTH IN CHARACTERS
          LA        A0,A15              LOAD CHARACTERS IN QUESTION
          DSL       A0,36               SHIFT FOR DIVIDE
          AA,U      A1,5                ADD FOR COVERED DIVIDE
          DI,U      A0,6                COMPUTE WORDS IN MESSAGE
          SA        A0,UQLW             SAVE WORD LENGTH OF USER QUESTION
          LR        R1,A0               LOAD WORDS IN QUESTION
          LA        A1,(1,UQUES)        LOAD USER QUESTION BUFFER ADDRESS
          LA        A0,(1,REPLY)        LOAD REPLY BUFFER POINTER
          BT        A1,,*A0             COPY MESSAGE TO SAVE AREA
          J         1,X4                RETURN TO NORMAL EXIT
.
.         IF A USER'S QUESTION WAS NOT PHRASED PROPERLY, INFORM
.         HIM OF THE PROPER ENGLISH SYNTAX FOR A QUESTION WHICH
.         DISTINGUISHES ONE ANIMAL FROM ANOTHER.
.
BADPREF   LX,U      X6,ASKRIGHT         LOAD CLARIFICATION STATEMENTS
          LMJ       A2,RANEDIT          EDIT ONE INTO THE IMAGE
          F$PRT     1                   PRINT THE CLARIFICATION
          J         0,X4                GET A NEW QUESTION FROM THE USER
/.
.
.         THE USER'S ANIMAL OCCURS ELSEWHERE IN THE TREE.  SCAN FROM
.         THE ANIMAL WE THOUGHT IT WAS AND THE FIND OF THE USER'S
.         ANIMAL IN THE TREE THROUGH THE BACK LINKS TO FIND THE
.         QUESTION AT WHICH THE USER DIVERGED FROM THE PATH WHICH
.         LEADS TO THE OCCURRENCE OF HIS ANIMAL.  WHEN WE FIND THE
.         QUESTION (AND WE BETTER!), SEVERAL INTERESTING ALTERNATIVES
.         PRESENT THEMSELVES: MORE ABOUT THIS LATER, AFTER WE FIND
.         IT.
.
.         X8 = LEAF NODE USER DISAGREED WITH
.         X5 = LEAF NODE USER'S ANIMAL WAS FOUND IN
.
ALRDUP    SX        X5,UFIND            SAVE FIND OF USER ANIMAL
          LA        A0,NODEBL+MEMORY,X5 A0 = QUESTION BEFORE OUR ANIMAL
          LA        A14,X5              A14 = PREVIOUS LINK
APRCPV    LA        A1,NODEBL+MEMORY,X8 A1 = QUESTION BEFORE USER'S ANIMAL
ALRCPN    TNE       A0,A1               HAVE WE FOUND THE COMMON QUESTION ?
          J         ALRFDV              YES.  WE HAVE FOUND THE DIVERGENT
.                                       QUESTION.  GO COGITATE ON IT.
          LA        A1,NODEBL,A1        NO.  CALL THE PREVIOUS QUESTION
          JNZ       A1,ALRCPN           LOOP IF NOT BASE OF TREE
          LA        A14,A0              SAVE PREVIOUS LINK
          LA        A0,NODEBL+MEMORY,A0 LINK TO PREVIOUS NODE
          JNZ       A0,ALRCPV           ALWAYS JUMPS.
          EABT$     .                   HEE, HEE, HEE.
.
.
.         WE GET HERE UPON FINDING THE QUESTION AT WHICH THE USER
.         DIVERGED FROM THE PATH WHICH LEADS TO THE OTHER OCCURRENCE
.         OF HIS ANIMAL IN THE TREE.  AT THIS POINT THERE ARE THREE
.         POSSIBILITIES FOR US TO CONSIDER.
.
.         1.  THE USER ERRED WHEN HE ANSWERED THIS QUESTION ON
.             THE WAY TO THE LEAF WITH WHICH HE DISAGREED.  IN
.             THIS CASE THE USER IS WRONG AND SHOULD BE CHAS-
.             TISED AND IGNORED.
.
.         2.  THE USER ANSWERED CORRECTLY, BUT THE PERSON WHO
.             ORIGINALLY ENTERED THE ANIMAL HE WANTS TO ENTER
.             INCORRECTLY ANSWERED THE DIVERGENT QUESTION.
.             IN THIS CASE WE WANT TO REMOVE THE ANIMAL
.             CURRENTLY IN OUR TREE (REMOVING THE QUESTION WHICH
.             PRECEDED IT) AND ENTER THE USER'S ANIMAL.
.
.         3.  THE QUESTION ON WHICH THE DIVERGENCE OCCURRED CAN
.             BE ANSWERED EITHER WAY FOR THE ANIMAL IN QUESTION.
.             FOR EXAMPLE, THE QUESTION 'DOES IT LIVE ON THE LAND'
.             COULD BE ANSWERED EITHER 'YES' OR 'NO' FOR A TURTLE.
.             IF THIS IS THE CASE, WE WANT TO ENTER THE USER'S
.             ANIMAL IN THE TREE, AND ALSO LEAVE THE ONE ALREADY
.             ENTERED IN PLACE.  WE WILL COUNT ON A SUBSEQUENT
.             REBALANCE TO CONSOLIDATE THE ANIMALS.
.
ALRFDV    SA        A1,A13              SAVE DIVERGENT NODE
          E$DIT     REPLPK              SET UP EDITOR ON REPLY LINE
          LX        X6,A13              LOAD DIVERGENT QUESTION NODE
          LA,U      A0,NODETEXT+MEMORY,X6 LOAD QUESTION TEXT ADDRESS
          LA        A1,NODELEN+MEMORY,X6 LOAD LENGTH IN CHARACTERS
          SA        A1,A15              SAVE QUESTION LENGTH FOR PLUGGEN
          E$COPY    .                   COPY QUESTION TO REPLY BUFFER
          LMJ       X7,PLUGGEN          TRY TO SUBSTITUTE USER ANIMAL
.                                       FOR PRONOUN IN THE QUESTION.
          J         NODVPL              CAN'T PLUG, CIRCUMLOCUTE.
          ON        MAINTENANCE
          TNZ       MAINT               IN MAINTENANCE MODE ?
          J         ALRMSK              NO.  SKIP ADDRESS EDITING
          SX        X8,A5               SAVE CURRENT NODE POINTER
          LX        X8,X6               LOAD ADDRESS OF QUESTION NODE
          LMJ       X4,RELADR           EDIT ADDRESS OF NODE
          LX        X8,A5               RESTORE NODE POINTER
ALRMSK    OFF       MAINTENANCE
          LMJ       X7,STICKEM          INSERT USER ANIMAL IN QUESTION
          LA        A0,(1,REPLY)        LOAD THE REPLY BUFFER ADDRESS
          LA        A1,(1,UQUES)        LOAD QUESTION BUFFER ADDRESS
          LR,U      R1,14               LOAD TEXT BUFFER LENGTH
          BT        A1,,*A0             SAVE PLUGGED TEXT IN USER QUESTION TEXT
ALRFPCS   LMJ       X4,DECIDE           REPOSE THIS QUESTION TO THE USER,
.                                       THIS TIME WITH HIS ANIMAL PLUGGED
.                                       IN FOR EXPLICITNESS.
          J         DVRYES              YES ANSWER.  CHECK THE YES LINK
.
.
.         NOW THAT WE HAVE THE USER'S ANSWER TO THE DIVERGENT QUESTION,
.         WE COMPARE HIS ANSWER TO THE ANSWER HE GAVE THEN THE QUESTION
.         WAS POSED ON THE ORIGINAL QUESTION AND ANSWER SESSION.  IF
.         THE ANSWER IS THE SAME, THE USER SEEMS WILLING TO STICK TO
.         HIS GUNS ON THE CONTENTION THAT HE ANSWERED CORRECTLY FOR
.         HIS ANIMAL.  IF THE USER IS SURE, WE WILL THEN TRY TO ASK HIM
.         WHETHER IT CAN BE BOTH WAYS FOR HIS ANIMAL.  BASED UPON THE
.         ANSWER TO THAT QUESTION, WE WILL EITHER DELETE THE OLD
.         ANIMAL OR LEAVE BOTH IT AND HIS NEW OCCURRENCE OF IT IN THE
.         MEMORY TREE.
.
          LA        A1,A13              LOAD DIVERGENT QUESTION NODE
          TNE       A14,NODENL+MEMORY,A1 DOES OUR FIND OF THE
.                                        ANIMAL IN THE TREE COME OFF THE 'NO'
.                                        LINK ?
          J         DVUERR              YES.  USER ERRED THE FIRST TIME
          J         DVCFRM              NO.  USER IS CONSISTENT.
.
.         'YES' ANSWER TO DIVERGENT QUESTION
.
DVRYES    LA        A1,A13              LOAD DIVERGENT QUESTION NODE
          TNE       A14,NODEYL+MEMORY,A1 DOES OUR FIND COME OFF
.                                        THE 'YES' LINK ?
          J         DVUERR              YES.  USER IS INCONSISTENT.  TELL HIM SO
.
.
.         THIS CODE IS ENTERED AFTER THE USER HAS CONFIRMED HIS ORIGINAL
.         ASSERTION THAT THE DIVERGENT QUESTION WAS PREVIOUSLY ANSWERED
.         CORRECTLY FOR HIS ANIMAL.  WE NOW SCAN THE PLUGGED QUESTION
.         AND TRY TO REFORMULATE IT INTO A QUESTION WHICH ASKS WHETHER
.         THE STATEMENT COULD BE EITHER TRUE OR FALSE FOR HIS ANIMAL.
.         BASED UPON HIS ANSWER, WE EITHER REMOVE THE OTHER OCCURRENCE
.         OF HIS ANIMAL OR LET IT STAND.
.
.         (THEN, OF COURSE, WE MUST MAKE SURE THERE ISN'T YET ANOTHER
.         INSTANCE OF HIS ANIMAL IN THE TREE).
.
DVCFRM    JZ        A15,DVCNFG          SKIP IF UNABLE TO PLUG QUESTION
          LA        A0,UQUES            LOAD FIRST WORD OF REPLY
          LA        A1,(1,0)            LOAD SEARCH POINTER
          LR,U      R1,CVFTL            LOAD LENGTH OF CAN/DOES TABLE
          SE        A0,CVFT,*A1         LOOK FOR WORD IN LEGAL TABLE
          J         DVCNFG              NOT FOUND.  MUST ASK THE GENERAL FORM
.
.         WE HAVE NOW ESTABLISHED THAT THE USER'S QUESTION BEGINS
.         WITH THE PROPER ENGLISH SYNTAX WHICH WILL PERMIT US TO
.         MODIFY IT INTO A QUESTION BY WHICH WE CAN ASK WHETHER THE
.         QUESTION COULD BE ANSWERED BOTH WAYS FOR THE ANIMAL IN
.         CONTENTION.  THE BASIC TRANSFORMATION IS AS FOLLOWS:
.
.         QUESTION: 'DOES IT LIVE ON THE LAND'
.         ANIMAL:   'A TURTLE'
.
.         NEW QUESTION:
.         'DOES <A TURTLE>[ BOTH] LIVE ON THE LAND[ AND NOT] LIVE ON THE LAND?'
.
          F$MSG     UQUES               COPY FIRST PART OF QUESTION
          LA,U      A0,UANML            LOAD USER'S ANIMAL ADDRESS
          LA        A1,UANL             LOAD LENGTH OF USER'S ANIMAL
          F$COPY    .                   COPY USER'S ANIMAL INTO QUESTION
          F$FD3     (' BOTH')           EDIT 'BOTH' INTO QUESTION
          F$MSGR    .                   SKIP SECOND FMSG$ STOP
          F$MSGR    .                   COPY TO END OF QUESTION
          F$FD4     (' AND NOT')        FILL IN 'AND NOT'
          F$COLN    .                   REMEMBER WHERE WE ARE
          SA        A0,A4               SAVE POSITION IN LINE
          F$MSG     UQUES               RE-EDIT FIRST PART OF QUESTION
          F$MSGR    .                   IGNORE SECOND FMSG$ STOP
          F$COL     A4,,W               POSITION BACK TO ORIGINAL LOCATION
          F$MSGR    .                   COPY THE REST OF THE QUESTION
DVBASK    LMJ       X4,DECIDE           ASK USER 'CAN IT BE BOTH'
          J         DVLVBOTH            YES.  LEAVE BOTH IN TREE
          LA        A0,UFIND            LOAD FIND OF USER'S ANIMAL
          LMJ       X4,DELANIMAL        DELETE THE BADLY PLACED OCCURRENCE
.                                       OF THE USER'S ANIMAL IN THE TREE
.
DVCSCAN   LX        X5,UFIND            RELOAD FIND POINTER
          LA        A15,UANL            RELOAD LENGTH OF USER'S ANIMAL
          J         ALRQUN              CONTINUE SCAN OF TREE
.
DVLVBOTH  LA,U      A0,1                LOAD A ONE
          SA        A0,NEEDRBAL         MARK REBALANCE NEEDED
          J         DVCSCAN             CONTINUE SCAN OF TREE FOR USER'S ANIMAL

.
.         ASK GENERIC QUESTION WHEN UNABLE TO PLUG INTO QUESTION IN TREE
.
NODVPL    LA        A1,A15              LOAD QUESTION LENGTH
          LA,U      A0,REPLY            LOAD REPLY BUFFER ADDRESS
          LA,U      A15                 INDICATE UNABLE TO PLUG
          F$COPY    .                   COPY THE QUESTION TO FDIT$'S LINE
          J         ALRFPCS             ASK GENERIC QUESTION OF USER
.
.         ASK GENERAL QUESTION TO DETERMINE WHETHER QUESTION MAY BE
.         ANSWERED BOTH WAYS FOR THE USER'S ANIMAL.  THIS HAPPENS
.         WHEN NO PRONOUN IF FOUND IN THE QUESTION OR THE SYNTAX OF THE
.         QUESTION DOES NOT LEND ITSELF TO TRANSLATION TO THE 'BOTH
.         WAYS' FORM.
.
DVCNFG    LX,U      X6,BOTHWAYS         LOAD 'YES AND NO BOTH' MESSAGES
          LMJ       A2,RANFDIT          EDIT INTO THE LINE
          LA,U      A0,UANML            LOAD USER ANIMAL ADDRESS
          LA        A1,UANL             LOAD LENGTH OF USER ANIMAL
          F$COPY    .                   COPY USER ANIMAL INTO QUESTION
          J         DVBASK              ASK USER WHETHER ANSWER CAN BE EITHER WAY
.
.
.         THE USER ERRED ON A QUESTION OR CHANGED HIS MIND BETWEEN THE
.         FIRST AND SECOND POSINGS OF THE DIVERGENT QUESTION.  CHIDE
.         HIM GENTLY FOR HIS GROSS INCOMPETENCE AND DON'T INSERT HIS
.         ANIMAL IN THE TREE.
.
DVUERR    PRINT$    MUYM,MUYML          'MAKE UP YOUR MIND.'
          J         AGAING              THIS GAME IS DONE
/.
.
.         TREE MANIPULATION FUNCTIONS
.
.
.         CREATE NODE
.
.         LA,U      A0,<LENGTH OF TEXT IN CHARACTERS>
.         LMJ       X11,MAKENODE
.         <RETURN>                      A1 = NODE ADDRESS (RELATIVE)
.
.         THIS SUBROUTINE ALLOCATES A MEMORY BUFFER FOR THE NODE,
.         EXPANDING THE MEMORY IF NECESSARY.  THE NODE IS CHAINED
.         INTO THE LIST OF NODES (LINKS NODEFL, NODEBL).  THE
.         TEXT LENGTH IN CHARACTERS (NODELEN) IS INITIALISED, AND
.         THE NODE CREATOR INFORMATION (NODEUID, NODEAC) IS FILLED
.         IN.  THE FIELDS NODEYL, NODENL, NODEBL, NODEREFC, AND
.         NODEBITS WILL BE ZEROED.
.
MAKENODE  SA        A0,A5               SAVE LENGTH IN CHARACTERS
          DSL       A0,36               SHIFT CHARACTER LENGTH FOR DIVIDE
          AA,U      A1,5                ROUND UP FOR COVERED DIVIDE
          DI,U      A0,6                COMPUTE NODE TEXT LENGTH IN WORDS
          LA        A1,HIGHUSE          LOAD ABSOLUTE ADDRESS OF NEW NODE
          AU,U      A1,NODETEXT,A0      ADD TOTAL LENGTH OF NEW NODE
          TG        A2,HIGHCORE         NEED MEMORY BE EXPANDED ?
          LMJ       X4,NUFF             YES.  ALLOCATE ANOTHER MEMORY BLOCK
          SA        A2.HIGHUSE          UPDATE NEXT AVAILABLE ADDRESS
          ANA,U     A1,MEMORY           MAKE NODE ADDRESS RELATIVE
.
.         INITIALISE FIELDS IN THE NEW NODE
.
          SZ        M(NODELNK),A1       CLEAR YES AND NO LINKS
          SA        A5,M(NODELEN),A1    SET TEXT LENGTH IN NODE
          SZ        M(NODEREFC),A1      ZERO REFERENCE COUNT ON NODE
          SZ        M(NODEBL),A1        CLEAR QUESTION LINK IN NODE
          SZ        M(NODEBITS),A1      CLEAR NODE TYPE BITS
.
.         ATTACH NODE TO LINEAR NODE LIST
.
          LA        A0,M(NODEBKL)       LOAD LINK TO LAST NODE IN LINK
          SA        A0,M(NODEBKL),A1    SET AS PREDECESSOR TO NEW NODE
          SA        A1,M(NODEBKL)       SET NEW NODE AS LAST
          SA        A1,M(NODEFL),A0     SET NEW NODE AS SUCCESSOR TO PREVIOUS
          SZ        M(NODEFL),A1        MARK HEAD AS SUCCESSOR TO NEW NODE
.
          LA        A0,USERID           LOAD USERID FOR CURRENT USER
          SA        A0,M(NODEUID),A1    SET USERID OF CREATOR
          DL        A4,ACCOUNT          LOAD ACCOUNT OF CREATOR
          DS        A4,M(NODEAC),A1     SET CREATOR'S ACCOUNT IN NODE
          J         0,X11               RETURN TO CALLER
.
.         DELETE ANIMAL
.
.         LA,U      A0,<ANIMAL NODE>    (RELATIVE)
.         LMJ       X11,DELANIMAL
.         <RETURN>
.
DELANIMAL TZ        M(NODELNK),A0       CALLED DELANIMAL ON A QUESTION ?
          ERR$      .                   YES.  BETTER TAKE A DUMP
          LA        A1,M(NODEBITS),A0   LOAD MODE BITS FOR NODE
          OR,U      A1,NBDEL            MARK THE NODE DELETED
          SA        A2,M(NODEBITS),A0   SET DELETE BIT IN NODE BITS
          LA        A1,M(NODEBL),A0     LOAD LINK TO PREVIOUS QUESTION
          LA        A3,M(NODEYL),A1     LOAD YES LINK FROM QUESTION
          TNE       A3,A0               IS YES LINK TO DELETED ANIMAL ?
          LA        A3,M(NODENL),A1     YES.  TAKE THE NO LINK
          TNE       A1,BASENODE         IS PREVIOUS QUESTION THE BASE NODE ?
          J         DELBN               YES.  WE WILL HAVE A NEW BASE NODE
          LA        A2,M(NODEBL),A1     LOAD NODE PRIOR TO QUESTION NODE
          TNE       A1,M(NODEYL),A2     WAS QUESTION OFF YES LINK ?
          SA        A3,M(NODEYL),A2     YES.  UPDATE YES LINK
          TNE       A1,M(NODENL),A2     WAS QUESTION OFF NO LINK ?
          SA        A3,M(NODENL),A2     YES.  ATTACH REMAINDER TO NO LINK
          SA        A2,M(NODEBL),A3     UPDATE BACK LINK IN REMAINING CHAIN
DELASF    LA        A2,M(NODEBITS),A1   LOAD BITS FROM QUESTION
          OR,U      A2,NBDEL            MARK THE QUESTION DELETED
          SA        A3,M(NODEBITS),A1   SET DELETE FLAG IN QUESTION NODE
          LA,U      A2,1                LOAD A ONE
          SA        A2,NEEDPACK         MARK 'TREE NEEDS TO BE PACKED'
          ON        MAINTENANCE
          TNZ       MAINT               IN MAINTENANCE MODE ?
          J         MMDAN               NO.  SKIP LOGGING OF DELETED QUESTIONS
          SX        X8,A5               SAVE CURRENT NODE POINTER
          LX        X8,A1               LOAD POINTER TO DELETED QUESTION
          F$MSG     ALSDEL              EDIT 'ALSO DELETED:  '
          SX        X4,A4               SAVE CALL ADDRESS
          LMJ       X4,RELADR           EDIT ADDRESS OF DELETED NODE
          LX        X4,A4               RELOAD THE CALL ADDRESS
          LA,U      A0,M(NODETEXT),X8   LOAD QUESTION TEXT ADDRESS
          LA        A1,M(NODELEN),X8    LOAD LENGTH OF QUESTION
          F$COPY    .                   COPY DELETED QUESTION
          LX        X8,A5               RELOAD CURRENT NODE POINTER
          F$CHAR    '?'                 EDIT QUESTION MARK
          F$PRT     1                   PRINT THE DELETED QUESTION
MMDAN     OFF       MAINTENANCE
          J         0,X4                RETURN TO CALLED
.
.         HANDLE DELETION OF BASE NODE
.
DELBN     SA        A3,BASENODE         SET REMAINDER CHAIN HEAD AS BASE NODE
          SZ        M(NODEBL),A3        MARK REMAINDER CHAIN WITH NO BACK LINK
          J         DELASF              GO SET DELETE FLAG IN OLD BASE NODE
.
.
.         DELETE SUBTREE
.
.         LA,U      A0,<BASE NODE OF SUBTREE>
.         LMJ       X4,DELTREE
.         <RETURN>
.
.         ALL NONZERO LINKS OF THE DESIGNATED SUBTREE WILL BE FOLLOWED
.         AND THE DELETE BIT FILL BE SET FOR EACH NODE ENCOUNTERED.
.
          ON        MAINTENANCE
DELTREE   LA        A1,A0               SET RUNNING POINTER TO BASE NODE
          SX        X4,A4               SAVE CALL ADDRESS
          SX        X8,A5               SAVE ORIGINAL X8
          SA        A0,A6               SAVE BASE NODE POINTER
DELTFL    TNZ       M(NODELNK),A1       IS THIS A LEAF NODE ?
          J         DELTLEAF            YES.  DELETE A LEAF NODE
          TZ        M(NODENL),A1        DOES NODE HAVE A 'NO' LINK ?
          LA        A1,M(NODENL),A1     YES.  ADVANCE UP NO LINK
          TZ        M(NODEYL),A1        DOES THIS (OR NEXT) NODE HAVE A
.                                       YES LINK ?
          LA        A1,M(NODEYL),A1     YES.  ADVANCE UP YES LINK
          J         DELTFL              LOOP LOOKING FOR END OF TREE
.
DELTLEAF  LA        A2,M(NODEBITS),A1   LOAD NODE STATUS BITS
          OR,U      A2,NBDEL            MARK THIS NODE DELETED
          SA        A3,M(NODEBITS),A1   SET DELETE BIT IN NODE
          TNE       A1,A6               JUST DELETED BASE NODE OF SUB-TREE
          J         DELTDONE            YES.  SUB TREE ALL DELETED
          F$MSG     ALSDEL              EDIT 'ALSO DELETED' MESSAGE
          LX        X8,A1               LOAD CURRENT NODE POINTER
          LMJ       X4,RELADR           EDIT RELATIVE ADDRESS OF NODE
          LA,U      A0,M(NODETEXT),X8   LOAD NODE TEXT ADDRESS
          LA        A1,M(NODELEN),X8    LOAD LENGTH OF NODE TEXT
          F$COPY    .                   COPY THE NODE TEXT
          F$PRT     1                   PRINT THE DELETED ITEM
          LA        A1,X8               RESTORE CURRENT NODE POINTER
          LA        A2,M(NODEBL),A1     LOAD LINK TO PREVIOUS NODE
          TNE       A1,M(NODENL),A2     CHAINED OFF 'NO' LINK ?
          SZ        M(NODENL),A2        YES.  CLEAR NO LINK
          TNE       A1,M(NODEYL),A2     CHAINED OFF 'YES' LINK ?
          SZ        M(NODEYL),A2        YES.  CLEAR THE YES LINK
          LA        A1,A2               LOAD ADDRESS OF PREVIOUS NODE
          J         DELTFL              LOOP DELETING SUB TREE
.
DELTDONE  LA,U      A1,1                LOAD A ONE
          SA        A1,NEEDPACK         MARK TREE NEEDS TO BE PACKED
          LX        X4,A4               RESTORE CALL ADDRESS
          LX        X8,A5               RELOAD CURRENT NODE POINTER
          J         0,X4                RETURN TO CALL
.
.
.         PACK DELETED NODE FROM TREE
.
.         LMJ       X4,PACK
.         <RETURN>
.
P         PROC      1                   ADJUST A LINK TO NEW ADDRESSING
ADJUST*   NAME      0
          LA        A3,P(1,1),P(1,2)    LOAD ADDRESS TO BE ADJUSTED
          TG        A3,X1               IS IT ABOVE DELETED NODE ?
          ANA       A3,A2               YES.  SUBTRACT LENGTH OF DELETED ITEM
          SA        A3,P(1,1),P(1,2)    STORE THE ADJUSTED ADDRESS BACK
          END       .
.
PACK      SZ        NEEDPACK            CLEAR PACK STILL NEEDED
PACK1     LX,U      X1                  CLEAR CURRENT NODE POINTER
.
PKNEXT    LX        X1,M(NODEFL),X1     LOAD LINK TO NEXT NODE
          TNZ       X1                  END OF TREE ?
          J         0,X4                YES.  PACK IS NOW COMPLETE
          LA        A0,M(NODEBITS),X1   LOAD STATUS BITS FOR THIS NODE
          TOP,U     A0,NBDEL            IS THIS NODE DELETED ?
          J         PKNEXT              NO.  LOOK AT THE NEXT NODE
          LA        A0,M(NODEFL),X1     LOAD LINK TO NEXT NODE
          LA        A1,M(NODEBKL),X1    LOAD LINK TO PREVIOUS NODE
          SA        A0,M(NODEFL),A1     ATTACH NEXT NODE TO PREVIOUS NODE
          SA        A1,M(NODEBKL),A0    ATTACH PREVIOUS NODE TO NEXT NODE
          LA        A2,M(NODELEN),X1    LOAD LENGTH OF DELETED NODE
          DSL       A2,36               SHIFT DOWN FOR DIVIDE
          AA,U      A3,5                ROUND UP FOR COVERED DIVIDE
          DI,U      A2,6                COMPUTE WORDS OF TEXT IN ITEM
          AA,U      A2,NODETEXT         ADD LENGTH OF HEADER
          LX        X2,M(NODEFL)        LOAD LINK TO FIRST NODE IN TREE
.
.         ADJUST ALL ADDRESSES ABOVE THE DELETED ITEM
.
PKADNX    TNZ       X2                  END OF TREE ?
          J         PKADJC              YES.  ADDRESS ADJUSTMENT COMPLETE
          LA        A4,M(NODEFL),X2     LOAD ORIGINAL FORWARD LINK FOR NODE
.                                       (ADJUSTMENT MAY CHANGE IT)
          ADJUST    M(NODEYL),X2        ADJUST YES LINK FROM NODE
          ADJUST    M(NODENL),X2        ADJUST THE NO LINK, ALSO
          ADJUST    M(NODEBL),X2        FIX THE QUESTION LINK
          ADJUST    M(NODEFL),X2        ADJUST THE FORWARD LINK
          ADJUST    M(NODEBKL),X2       ADJUST THE BACKWARD LINK
          LX        X2,A4               LOAD LINK TO THE NEXT NODE
          J         PKADNX              GO ADJUST THE NEXT NODE
.
PKADJD    ADJUST    BASENODE            FIX THE BASE NODE ADDRESS
          ADJUST    M(NODEFL)           FIX THE HEAD FORWARD LINK
          ADJUST    M(NODEBKL)          FIX THE LINK TO THE LAST NODE
          LA        A0,(1,0)            LOAD POINTER TO COMPRESS THE TREE
          AA,U      A0,,X1              COMPUTE OFFSET OF DELETED NODE
          AU,U      A0,,A2              START OF TREE ABOVE NODE TO BE COMPRESSED
          LA        A3,HIGHUSE          LOAD HIGHEST WORD IN USE
          ANA,U     A3,MEMORY,A1        A3 = NUMBER OF WORDS ABOVE THE GAP
          SA        A3,R1               SET AS COUNT FOR TREE COMPRESSION
          BT        A0,MEMORY,*A1       SQUEEZE THE DELETED ITEM OUT OF THE TREE
          ANA       A2,HIGHUSE          COMPUTE -(HIGHEST WORD IN USE)
          SNA       A2,HIGHUSE          UPDATE HIGHEST WORD IN USE
          J         PACK1               LOOK FOR MORE DELETED NODES
/.
.
.
.         ASK THE USER A QUESTION
.
.         LX,U      X6,<QUESTION CONTROL BLOCK>
.         LMJ       X5,QUESTION
.         A15 = LENGTH OF ANSWER IN CHARACTERS
.
QUESTION  LMJ       A2,RANFDIT          EDIT A RANDOM QUESTION FROM BLOCK
QUESTK    F$FD2     ('? ')              APPEND QUESTION MARK
          F$CHAR    EOL                 APPEND END OF LINE
          TREAD$P   GIGO.               GET THE ANSWER
          F$DT      .                   CLEAR THE QUESTION LINE
          TNZ,U     0,A0                NULL ANSWER ?
          J         QUESTION            GO ASK ANOTHER QUESTION
QUESTL    LA,U      A0,,A0              GET RID OF H1
          LA        A2,REPLY-1,A0       LOAD LAST WORD IN REPLY
          LA,U      A15,,A0             LOAD WORDS RETURNED
          MSI,U     A15,6               COMPUTE CHARACTERS IN FULL QUESTION
          LR,U      R1,5                LOAD LOOP COUNTER
QSL       DSC       A1,6                SHIFT CHARACTERS FROM RIGHT INTO A1
          SSL       A1,30               RIGHT JUSTIFY CHARACTER
          TE,U      A1,' '              TRAILING SPACE ?
          J         0,X5                NO.  DONE SCANNING REPLY
          ANA,U     A15,1               YES.  DECREMENT LENGTH OF REPLY
          JGD       R1,QSL              LOOP FOR ALL CHARACTERS IN LAST WORD
          J         0,X5                RETURN TO CALLER
.
.
.         EDIT A RANDOM STRING FROM A QUESTION CONTROL BLOCK
.
.         LX,U      X6,<QUESTION CONTROL BLOCK>
.         LMJ       A2,RANFDIT
.         <RETURN>                      TEXT EDITED INTO FDIT IMAGE
.
RANFDIT   LA,U      A1                  CLEAR INITIAL QUESTION INDEX
          TNZ,H1    0,X6                MORE THAN ONE QUESTION IN BLOCK ?
          J         RFDSIMP             NO.  SIMPLE CASE
          TIME$     .                   GET RANDOM TIME
          DSL       A0,36               MOVE TIME DOWN TO A1
          DI,H1     A0,,X6              DIVIDE BY NUMBER OF QUESTIONS
RFDSIMP   AA,U      A1,,X6              COMPUTE ADDRESS OF QUESTION POINTER
          F$MSG     0,A1,H2             EDIT THE QUESTION TEXT
          J         0,A2                RETURN TO CALL
.
.
.         DECIDE A YES OR NO QUESTION
.
DECIDE    LX,U      X6,YORN             GET YES OR NO LIST FOR RETRY
          LMJ       X5,QUESTK           ASK THE ORIGINAL QUESTION
DECID1    LA,S1     A0,REPLY            LOAD FIRST CHARACTER OF REPLY
          TNE,U     A0,'Y'              YES ?
          J         0,X4                YES.  TAKE AFFIRMATIVE EXIT
          TNE,U     A0,'N'              NEGATIVE ?
          J         1,X4                YES.  TAKE NEGATIVE EXIT
.
.         LOOK FOR AFFIRMATIVE AND NEGATIVE SOUNDING WORDS IN A
.         TABLE.  CURRENTLY THIS TABLE IS ASSEMBLED IN, BUT IT
.         MAY BE KEPT IN THE MEMORY IN THE FUTURE.
.
          LA        A0,REPLY            LOAD USER'S REPLY TO THE QUESTION
          ON        MAINTENANCE
          TNE       A0,('THAT''S')      IS THE REPLY 'THAT'S IT' ?
          TNZ       MAINT               YES.  ARE WE IN MAINTENANCE MODE ?
          J         DECNOM              NO.  SKIP NODE FINDING CODE
          LX        X11,R5              YES.  LOAD CALL ADDRESS FOR FINDNODE
          TZ        X11                 WERE WE INVOKED BY FINDNODE ?
          J         0,X11               YES.  TAKE FINDNODE RETURN
DECNOM    OFF       MAINTENANCE
          LR,U      R5,YESLL            LOAD LENGTH OF 'YES' LIST
          LA        A1,(1,0)            LOAD SEARCH POINTER
          SE        A0,YESL,*A1         LOOK FOR ANSWER IN THE LIST
          J         $+2                 NOT FOUND.  CHECK THE 'NO' LIST
          J         0,X4                FOUND IN YES LIST.  TAKE YES REPLY
          LA        A1,(1,0)            RESTORE SEARCH POINTER
          LR,U      R1,NOLL             LOAD LENGTH OF 'NO' LIST
          SE        A0,NOL,*A1          LOOK FOR ANSWER IN 'NO' LIST
          J         $+2                 NOT FOUND.  REASK QUESTION
          J         1,X4                FOUND IN 'NO' LIST.  TAKE NO REPLY
          LMJ       X5,QUESTION         ASK THE QUESTION AGAIN
          J         DECID1              GO INTERPRET NEXT ANSWER
.
.
.         THIS SUBROUTINE SCANS A QUESTION FOUND IN THE BUFFER 'REPLY'
.         WHOSE LENGTH IS SPECIFIED BY A15 FOR A PRONOUN.  UPON FINDING
.         ONE, IT IS PLUGGED WITH AN EMSG$ STOP CHARACTER TO PERMIT
.         SUBSTITUTION FOR IT.
.
.         LA,U      A15,<LENGTH OF QUESTION>
.         LMJ       X7,PLUGGEN
.         <NO PRONOUN>                  REPLY BUFFER UNCHANGED
.         <PRONOUN PLUGGED>
.
PLUGGEN   E$COL     0                   TAB TO START OF MESSAGE
.
.         SEARCH THE USER'S QUESTION FOR 'IT', AND SET UP TO
.         PLUG THE ANIMAL NAME WHEN WE ASK IT BACK TO HIM.
.
FINDIT    U$CHAR    .                   LOAD A CHARACTER FROM THE MESSAGE
          TE,U      A0,'I'              IS IT AN 'I' ?
          J         FIND1               NO.  ADVANCE TO NEXT START OF WORD
          U$CHAR    .                   LOAD NEXT CHARACTER AND PEED AHEAD
          TNE,U     A0,'T'              IS IT AN 'IT' ?
          TE,U      A2,' '              (MUST BE FOLLOWED BY SPACE)
          J         FIND2               NO.  RECOVER AND CONTINUE
          E$SKIP    -2                  BACK UP OVER THE 'I'
          E$FD1     ('&&')              EDIT TWO STOP CHARACTERS INTO THE LINE
          E$COL     A15,,W              TAB TO END OF MESSAGE
          E$CHAR    '&'                 INSERT END OF MESSAGE THERE
          E$DITX    .                   LEAVE EDITING MODE
          J         1,X7                PRONOUN PLUGGED.  RETURN
.
FIND2     TNE,U     A0,' '              WAS THE BAD CHARACTER A SPACE ?
          J         FIND3               YES.  SKIP ADVANCE TO NEXT SPACE
FIND1     U$POS1    ' '                 FIND THE END OF THIS WORD
          JN        A0,,X7              TAKE NO FIND RETURN IF NO PRONOUN
FIND3     U$POS3    .                   FIND NEXT NON-BLANK
          E$COLN    .                   RETURN COLUMN POINTER
          TG        A0,A15              BEYOND END OF USER IMAGE ?
          J         0,X7                YES.  NO PRONOUN IN SENTENCE
          J         FINDIT              GO PROCESS THIS CHARACTER
.
.
.         THIS SUBROUTINE INSERTS THE USER'S ANIMAL (FROM UANML)
.         INTO A QUESTION IN THE BUFFER 'REPLY' PREVIOUS SCANNED
.         BY 'PLUGGEN'.
.
.         LMJ       X7,STICKEM
.         <RETURN>                      QUESTION EDITED INTO FL$
.
STICKEM   F$MSG     REPLY               EDIT THE USER'S QUESTION BACK TO HIM
          LA,U      A0,UANML            LOAD ADDRESS OF USER'S ANIMAL
          LA        A1,UANL             LOAD LENGTH OF USER ANIMAL
          F$COPY    .                   COPY TEXT OF MESSAGE
          F$MSGR    .                   IGNORE DROPPED CHARACTERS
          F$MSGR    ,                   COPY REST OF QUESTION
          J         0,X7                RETURN TO CALLER
/.
.
.         INSTALL THE ANIMAL FILE AT A NEW SITE
.
INSTALL   F$DT      ,                   CLEAR THE EDITING LINE
          F$MSG1    CREMEM              EDIT THE @CAT IMAGE FOR THE MEMORY FILE
          F$MSG     MEMFILE             EDIT THE MEMORY FILE NAME
          F$MSGR    .                   ...WITH KEYS
          F$MSG1    CREMEM1             EDIT THE ASSIGN PARAMETERS
          CSF$      FL$                 GO TRY TO @CAT THE MEMORY FILE
          JN        A0,LATER            ERROR OFF IF UNASSIGNABLE
          F$DT      .                   CLEAR THE EDITING LINE
          F$MSG     USEMEM              COPY @USE IMAGE TO LINE
          F$MSG1    MEMFILE             EDIT NAME OF MEMORY FILE
          CSF$      FL$                 ATTACH @USE NAME TO MEMORY FILE
          F$DT      .                   CLEAR THE LINE
          SR        R15,LUPTIME         INITIALISE TIME OF LAST UPDATE
          IOW$      IOPW                INITIALISE THE MEMORY FILE
          TZ,S1     IOPW+3              NORMAL WRITE STATUS ?
          J         LATER               NO.  MAKE USER GO AWAY
          CSF$      FREEMEM             RELEASE THE MEMORY FILE
          J         ASGMA               GO RETRY THE ASSIGN
.
.         REWRITE MEMORY TO FILE WHEN USER IS DONE
.
AGAINQ    .
          ON        MAINTENANCE
          TZ        MAINT               MAINTENANCE MODE ?
          J         MAINTCMD            YES.  RETURN TO SCANNER AT END OF GAME
          OFF       MAINTENANCE
          LX,U      X6,PLAYAGAIN        LOAD QUESTIONS FOR PLAYAGAIN
          LMJ       A2,RANFDIT          EDIT A QUESTION TO ASK PLAY AGAIN ?
          LMJ       X4,DECIDE           DOES USER WANT TO PLAY AGAIN ?
          J         RESTART             YES.  GO START ALL OVER
.
RWOUT     TZ        NEEDPACK            DOES TREE NEED TO BE CONDENSED ?
          LMJ       X4,PACK             YES.  SQUEEZE OUT THE DELETED ITEMS
          LA        A0,HIGHUSE          LOAD HIGHEST ADDRESS ASSIGNED
          ANA,U     A0,MEMORY           SUBTRACT MEMORY START ADDRESS
          SA        A0,MEMLEN           SET LENGTH IN FILE
          SA,H1     A0,IOPW+4           SET LENGTH IN ACCESS WORD
          SR        R15,LUPTIME         SAVE TIME OF LAST UPDATE
          LA        A0,NUPDATE          LOAD THE NUMBER OF FILE UPDATES
          AA,U      A0,1                INCREMENT IT FOR THIS ONE
          SA        A0,NUPDATE          SAVE NUMBER OF FILE UPDATES
          IOW$      IOPW                REWRITE MEMORY TO FILE
          TZ,S1     IOPW+3              NORMAL STATUS ?
          EABT$     .                   ********
          J         EOFANS              SAY GOODBYE TO USER
.
.         EXPAND MEMORY WHEN REQUIRED
.
NUFF      LA        A0,HIGHCORE         LOAD HIGHEST ADDRESS
          AA,U      A0,512              INCREMENT BY ONE MEMORY BLOCK
          SA        A0,HIGHCORE         UPDATE HIGHEST AVAILABLE ADDRESS
          MCORE$    .                   ALLOCATE ONE MORE MEMORY BLOCK
          J         0,X4                RETURN AFTER ALLOCATION COMPLETE
/.
.
.         ANIMAL TREE MAINTENANCE PROCESSOR
.
          ON        MAINTENANCE
.
MAINTMAIN F$MSG     MAINTSO             EDIT MAINTENANCE SIGN-ON LINE
          F$FD1     (LEVEL)             EDIT LEVEL OF ANIMAL PROCESSOR
          ON        LOCLVL
          F$FD1     (LOCLVL)            EDIT LOCAL LEVEL
          OFF       LOCLVL
          F$MSGR    .                   COPY REST OF SIGN ON
          F$PRT     1                   PRINT THE MESSAGE
.
.         RETURN HERE TO FETCH THE NEXT COMMAND
.
MAINTCMD  F$CHAR    '*'                 EDIT A SOLICITATION
          F$CHAR    EOL                 EDIT LINE TERMINATOR
MCMA      TREAD$P   GIGO.               SOLICIT NEXT COMMAND FROM USER
          TNZ,U     0,A0                NULL RESPONSE BY USER ?
          J         MCMA                YES.  ASK HIM AGAIN
          LR,U      R5                  CLEAR JUMP-BACK ADDRESS FOR PLAY
          F$DT      .                   CLEAR THE EDITING LINE
.
.         SEARCH FOR THE COMMAND IN THE COMMAND TABLE
.
          LA,H1     A0,REPLY            LOAD FIRST THREE LETTERS OF COMMAND
          LR,U      R1,TMPCTL           LOAD LENGTH OF COMMAND TABLE
          LA        A1,(1,0)            LOAD COMMAND TABLE POINTER
          SE,H1     A0,TMPCT,*A1        LOOK FOR COMMAND IN TABLE
          J         MCMBAD              ILLEGAL COMMAND.  REJECT IT
          LA,H2     A0,TMPCT-1,A1       LOAD COMMAND ROUTINE ADDRESS
          J         0,A0                ENTER COMMAND ROUTINE
.
.         REJECT ILLEGAL COMMAND
```

```
.
MCMBAD    F$MSG     BADCMM              EDIT BAD COMMAND MESSAGE
          F$PRT     1                   PRINT ERROR MESSAGE
          J         MAINTCMD            ASK FOR ANOTHER COMMAND
.
.
.         CHANGE ANIMAL COMMAND (CA)
.
CHGANML   LMJ       X11,FINDNNODE       FIND ANIMAL TO BE CHANGED
          TZ        M(NODELNK),X8       IS SELECTED NODE AN ANIMAL ?
          J         NOTANML             NO.  GIVE ERROR MESSAGE
          LX,U      X6,NEWANML          LOAD QUERY FOR NEW ANIMAL
          LMJ       X5,QUESTION         ASK FOR THE NEW ANIMAL NAME
          LMJ       X5,SCANANI          SCAN THE SUBMITTED ANIMAL
.         LMJ       X11,DUPCHECK        CHECK FOR DUPLICATE ANIMAL
          LA        A0,UANL             LOAD LENGTH OF ANIMAL
          LMJ       X11,MAKENODE        ALLOCATE A NODE FOR THE ANIMAL
          LA        A0,M(NODEBL),X8     LOAD LINK TO QUESTION BEFORE
.                                       THE OLD ANIMAL NODE.
          LA,U      A4,,X8              LOAD ADDRESS OF OLD NODE
          TNE       A4,M(NODEYL),A0     CHAINED OFF YES LINK ?
          SA        A1,M(NODEYL),A0     YES.  ATTACH NEW NODE TO YES LINK
          TNE       A4,M(NODENL),A0     CHAINED OFF NO LINK ?
          SA        A1,M(NODENL),A0     YES.  ATTACH NEW NODE TO NO LINK
          SA        A0,M(NODEBL),A1     SET BACK LINK IN REPLACEMENT NODE
          LR        R1,UANLW            LOAD NEW ANIMAL LENGTH IN WORDS
          LA        A0,(1,0)            LOAD POINTER FOR MOVE
          AA,U      A0,M(NODETEXT),A1   FORM DESTINATION POINTER
          LX        X11,(1,UANML)       LOAD SOURCE POINTER
          BT        A0,,*X11            COPY TEXT TO NEW NODE
          LA        A1,M(NODEBITS),X8   LOAD BITS FROM OLD NODE
          OR,U      A1,NBDEL            SET DELETE BIT IN THE NODE
          SA        A2,M(NODEBITS),X8   UPDATE BITS IN THE NODE
          LA,U      A1,1                LOAD A ONE
          SA        A1,NEEDPACK         MARK TREE IN NEED OF PACK
          J         MAINTCMD            PROCESS THE NEXT COMMAND
.
NOTANML   F$MSG     NOTANMM             EDIT 'NOT ANIMAL' MESSAGE
          F$PRT     1                   PRINT THE MESSAGE
          J         MAINTCMD            GET THE NEXT COMMAND
.
.
.         CHANGE QUESTION COMMAND (CQ)
.
.
CHGQUES   LMJ       X11,FINDNODE        LOOK FOR NODE TO BE CHANGED
          TNZ       M(NODELNK),X8       IS IT A QUESTION NODE ?
          J         NOTQUES             NO.  GIVE ERROR MESSAGE
CQRA      LX,U      X6,REPLQUES         LOAD QUERY FOR NEW QUESTION
          LMJ       X5,QUESTION         ASK FOR REPLACEMENT QUESTION
          LMJ       X4,SCANQUES         SCAN THE REPLACEMENT QUESTION
          J         CQRA                ASK FOR QUESTION AGAIN IF BAD
          LA        A0,UQL              LOAD LENGTH OF QUESTION
          LMJ       X11,MAKENODE        ALLOCATE A NODE FOR QUESTION
          LA        A2,X8               LOAD THE OLD NODE ADDRESS
          LA        A0,M(NODELNK),X8    LOAD YES AND NO LINKS OF OLD QUESTION
          SA        A0,M(NODELNK),A1    SET LINKS IN REPLACEMENT QUESTION
          TNE       A2,BASENODE         ARE WE REPLACING THE BASE NODE ?
          J         CHGBSN              YES.  HANDLE SPECIALLY
          LA        A0,M(NODEBL),X8     LOAD LINK TO PREVIOUS QUESTION
          SA        A0,M(NODEBL),A1     SET BACK LINK IN NEW QUESTION
          TNE       A2,M(NODEYL),A0     CHAINED OFF YES LINK ?
          SA        A1,M(NODEYL),A0     YES.  ATTACH QUESTION TO YES LINK
          TNE       A2,M(NODENL),A0     CHAINED OFF THE NO LINK ?
          SA        A1,M(NODENL),A0     YES.  ATTACH TO THE NO LINK
CHQSTD    LR        R1,UQLW             LOAD QUESTION LENGTH IN WORDS
          LX        X11,(1,UQUES)       LOAD POINTER TO USER QUESTION
          LA,U      A0,M(NODETEXT),A1   LOAD NODE TEXT POINTER
          AA        A0,(1,0)            FORM POINTER TO TEXT
          BT        A0,,*X11            COPY TEXT TO NODE
          LA        A0,M(NODEBITS),X8   LOAD STATUS BITS FOR NODE
          OR,U      A0,NBDEL            MARK OLD NODE DELETED
          SA        A1,M(NODEBITS),A0   SET DELETE STATUS IN NODE
          LA,U      A0,1                LOAD A ONE
          SA        A0,NEEDPACK         MARK TREE NEEDS A PACK
          J         MAINTCMD            RETURN FOR NEXT COMMAND
.
CHGBSN    SA        A1,BASENODE         SET NEW QUESTION AS BASE NODE
          SZ        M(NODEBL),A1        SET BACK LINK IN NODE TO ZERO
          J         CHQSTD              RETURN TO COPY TEXT
.
.
.         DELETE ANIMAL COMMAND (DA)
.
DELANML   LMJ       X11,FINDNODE        LOCATE NODE TO BE DELETED
          TZ        M(NODELNK),X8       IS THIS AN ANIMAL NODE ?
          J         NOTANML             NO.  GIVE ERROR MESSAGE
          LA        A0,X8               LOAD NODE TO BE DELETED
          LMJ       X11,DELANIMAL       DELETE THE ANIMAL FROM THE TREE
          J         MAINTCMD            PROCESS NEXT COMMAND
.
NOTQUES   F$MSG     NOTQUEM             EDIT 'NOT QUESTION' MESSAGE
          F$PRT     1                   PRINT ERROR MESSAGE
          J         MAINTCMD            IGNORE THE COMMAND
.
.
.         DELETE QUESTION COMMAND (DQ)
.
DELQUES   LMJ       X11,FINDNODE        LOCATE THE QUESTION TO BE DELETED
          TNZ       M(NODELNK),X8       IS THE NODE A QUESTION ?
          J         NOTQUES             NO.  GIVE ERROR MESSAGE
          F$MSG     DELQASM             ASK WHICH PATH USER WANTS TO TAKE
          LMJ       X4,DECIDE           GET THE USER'S ANSWER
          J         DELQYES             YES ANSWER.  SAVE 'YES' LINK SUB TREE
          LA        A2,M(NODENL),X8     LOAD SUB-TREE CHAINED OFF 'NO' LINK
          SZ        M(NODENL),X8        MARK NO LINK SUBTREE REMOVED
          J         DELQDTR             GO REHOOK AND DELETE SUBTREE
.
DELQYES   LA        A2,M(NODEYL),X8     LOAD YES LINK SUBTREE
          SZ        M(NODEYL),X8        MARK 'YES' SUBTREE REMOVED
.
DLQDTR    LA        A0,X8               LOAD ADDRESS OF QUESTION NODE
          TNE       A0,BASENODE         IS THIS THE BASE NODE ?
          J         DLQBSN              YES.  PROCESS SPECIALLY
          LA        A1,M(NODEBL),X8     LOAD LINK TO PREVIOUS QUESTION
          SA        A1,M(NODEBL),A2     SET BACK LINK IN REMAINDER
          TNE       A0,M(NODEYL),A1     WAS THIS OFF YES LINK ?
          SA        A2,M(NODEYL),A1     YES.  CHAIN SUBTREE TO YES LINK
          TNE       A0,M(NODENL),A1     OR WAS IT OFF NO LINK OF PREDECESSOR ?
          SA        A2,M(NODENL),A1     YES.  ATTACH SUBTREE TO NO LINK
DLQSTD    LMJ       X4,DELTREE          DELETE THE SUBTREE NOT SELECTED
          J         MAINTCMD            RETURN FOR NEXT COMMAND
.
DLQBSN    SA        A2,BASENODE         ATTACH SUBTREE AS BASE NODE
```

```
          SZ        M(NODEBL),A2        MARK REMAINDER CHAIN WITH NO BACK LINK
          J         DLQSTD              GO DELETE THE OTHER SUBTREE
.
.
.         LIST TREE COMMAND (LT)
.
LISTREE   PLINE$    0                   SKIP TO TOP OF PAGE
          F$MSG     LTMTX               EDIT LIST TREE HEADER MESSAGE
          F$FD1     FILELEVEL           EDIT LEVEL OF MEMORY FILE
          F$FD1     FILELOCL            EDIT LOCAL LEVEL OF MEMORY FILE
          F$MSGR    .                   COPY TO LAST UPDATE TIME
          F$DAY1    LUPTIME             EDIT DATE OF LAST UPDATE
          F$MSGR    .                   COPY TO TIME
          F$TIME    LUPTIME             EDIT TIME OF LAST UPDATE
          F$MSGR    .                   COPY TO UPDATE SERIAL NUMBER
          F$DECV    NUPDATE             EDIT NUMBER OF FILE UPDATES
          F$MSGR    .                   COPY THE REST OF THE MESSAGE
          F$PRT     1                   PRINT THE FILE LISTING HEADER, LINE 1.
          F$MSG     LTMTX1              EDIT SECOND LINE OF FILE LIST HEADER
          F$DECV    MEMLEN              EDIT LENGTH OF MEMORY FILE
          F$MSGR    .                   COPY TO BASE NODE
          F$DECV    BASENODE            EDIT ADDRESS OF ROOT NODE
          F$MSGR    .                   COPY REST OF HEADER LINE
          F$PRT     1                   PRINT SECOND LINE OF HEADER
          F$PRT     2                   SKIP BEFORE COMMENCING NODE LISTING
          LX        X5,M(NODEFL)        LOAD LINK TO FIRST NODE
.
.         EDIT NODES FROM THE TREE
.
LTNEXN    TNZ       X5                  END OF NODES IN TREE ?
          J         LTDONE              YES.  COMPLETE LISTING
          F$DECF    5,0,X5,U            EDIT RELATIVE ADDRESS OF THIS NODE
          F$CHAR    ':'                 EDIT A COLON AFTER IT
          F$SHIP    2                   SKIP AFTER THE NODE ADDRESS
          LA        A8,M(NODELEN),X5    LOAD LENGTH OF NODE TEXT
          LA,U      A0,M(NODETEXT),X5   LOAD NODE TEXT LENGTH
          SA        A0,A9               SAVE FOR POSSIBLE CONTINUATION
          LA        A1,A8               LOAD LENGTH TO EDIT
          TG,U      A1,A8+1             TOO MUCH TO FIT ON ALLOTTED SPACE
          LA,U      A1,48               YES.  TRIM TO FIELD LENGTH
          ANA       A8,A1               DECREMENT LENGTH LEFT TO EDIT
          F$COPY    .                   COPY UP TO 48 CHARACTERS TO LINE
          AA,U      A9,8                INCREMENT ADDRESS FOR NEXT LINE
          F$COL     57                  TAB TO LINKS COLUMN
          TNZ       M(NODELNK),X5       IS THIS A LEAF NODE ?
          J         LTZER1              YES.  BLANK LINKS WHEN ZERO
          F$DECF    5,M(NODEYL),X5      EDIT THE YES LINK FROM NODE
          F$SKIP    1                   SKIP BEFORE 'NO' LINK
          F$DECF    5,M(NODENL),X5      EDIT NO LINK FOR QUESTION NODE
          F$SKIP    1                   SKIP AFTER THE NO LINK
          J         LTLBL               GO EDIT THE BACK (QUESTION) LINK
.
LTZERL    F$SKIP    12                  SKIP SPACE WHERE LINKS WENT
LTLBL     F$DECF    5,M(NODEBL),X5      EDIT QUESTION LINK FOR NODE
          F$SKIP    1                   SKIP BEFORE NODEBITS FIELD
          TNZ       M(NODEBITS),X5      ARE ALL STATUS BITS ZERO ?
          J         LTNBZR              YES.  BLANK THE FIELD
          F$OCTF    6,M(NODEBITS),X5    EDIT THE NODE BITS
          F$SKIP    1                   SKIP AFTER THE BITS
          J         LTNBNZR             GO EDIT THE REFERENCE COUNT
.
LTNBZR    F$SKIP    7                   SKIP NODE BITS FIELD
```

```
LTNBNZR   F$DECF    6,M(NODEREFC),X5    EDIT REFERENCE COUNT FOR NODE
          F$SKIP    1                   SKIP AFTER REFERENCE COUNT
          F$COPY    6,M(NODEUID),X5     EDIT USERID OF NODE'S CREATOR
          F$SKIP    1                   SKIP AFTER USERID
          F$COPY    12,M(NODEAC),X5     EDIT ACCOUNT OF NODE'S CREATOR
          F$PRT     1                   PRINT THE NODE LISTING
          LX        X5,M(NODEFL),X5     ADVANCE TO NEXT NODE IN TREE
LTBCTD    JZ        A8,LTNEXN           EDIT NEXT NODE IF ALL TEXT EDITED
.
.         EDIT CONTINUATION OF NODE TEXT TOO LONG TO FIT IN ONE
.         LINE OF STANDARD NODE LISTING.  AS MANY CONTINUATION
.         LINES AS ARE REQUIRED WILL BE EDITED.
.
          F$SKIP    9                   SKIP THE NODE ADDRESS FIELD
          F$FD1     ('...')             INDICATE THIS IS A CONTINUATION
          LA        A0,A9               LOAD ADDRESS OF NODE TEXT
          LA        A1,A8               LOAD CHARACTERS LEFT TO EDIT
          TG,U      A1,48+1             STILL TOO MUCH TO FIT ?
          LA,U      A1,48               YES.  LIMIT TO FIELD SIZE
          ANA       A8,A1               COMPUTE CHARACTERS LEFT TO EDIT
          AA,U      A9,8                ADVANCE ADDRESS OF TEXT
          F$COPY    .                   COPY TEXT DATE INTO CONTINUATION
          F$PRT     1                   PRINT CONTINUATION TEXT
          J         LTBCTD              EDIT MORE CONTINUATIONS IF NEEDED
.
.         TREE LISTING COMPLETE
.
LTDONE    PLINE$    0                   EJECT PAGE AT END OF LISTING
          J         MAINTCMD            GO GET ANOTHER COMMAND
.
.
.         LOCATE A NODE FOR MAINTENANCE
.
.         THE USER PLAYS THE GAME.  UPON REACHING THE DESIRED NODE,
.         HE ANSWERS 'THAT'S IT' TO THE QUESTION.
.
FINDNODE  SX        X11,R5              SAVE RETURN ADDRESS FROM FINDNODE
          J         RESTART             GO LOOK FOR THE NODE
.
.
.         EDIT RELATIVE ADDRESS OF NODE
.
.         LX,U      X8,<RELATIVE ADDRESS OF NODE>
.         LMJ       X4,RELADR
.
RELADR    F$CHAR    '['                 EDIT A LEFT BRACKET
          F$DECV    0,X8,U              EDIT THE RELATIVE ADDRESS
          F$CHAR    ']'                 EDIT THE CLOSING BRACKET
          F$SKIP    1                   SKIP AFTER THE BRACKET
          J         0,X4                RETURN TO CALLER
.
          OFF       MAINTENANCE
/.
.
.         MEMORY BUFFER DEFINITION
.
.         THIS STRUCTURE MUST CONTAIN ALL INFORMATION TO BE REMEMBERED
.         FROM EXECUTION TO EXECUTION.  THE HEADING OF THIS FILE IS READ
.         IN EACH TIME TO RETRIEVE THE CURRENT STATE OF THE MEMORY FILE.
.
P         PROC      *3                  YES,NO  LEN   BACKLINK,BITS
NODE*     NAME      0
```

```
ND$*(A(0)) EQU      $-MEMORY            SAVE NODE INDEX AND ADDRESS
A*(0)     EQU       A(0)+1              UPDATE NODE COUNT
*         EQU       $-MEMORY            MAKE TAG RELATIVE
          *         P(1,1),P(1,2)       YES AND NO LINKS
F1        FORM      6,12,18
          F1        P(2,1)              LENGTH IN CHARACTERS
          *         P(3,1),P(3,2)       BACKLINK AND BITS
          *         ND$(A(0)-2),ND$(A(0)) FORWARD AND BACK LINKS
          '*ORIG*'                      ORIGINAL NODE
          'INSTALLATION'                INSTALLED AT START
          END
A(0)      EQU       0                   RESET NODE COUNT TO ZERO
$(4).
MEMORY    .
          'ANIMAL'                      SENTINEL
MEMLEN    *         MEML                MEMORY LENGTH IN WORDS
BASENODE  *         LAND                BASE NODE ADDRESS
NODECHAIN *         ND$(ND$-1),ND$(0)   LINKED LIST OF NODES
FILELEVEL *         LEVEL               LEVEL OF MEMORY FILE
FILELOCL  *         LOCLVL              MEMORY FILE LOCAL LEVEL
LUPTIME   *         $-$                 TIME AND DATE OF LAST UPDATE (TDATE$)
NUPDATE   *         0                   NUMBER OF FILE UPDATES
.
          RES       28-($-MEMORY)       ADJUST TO SECTOR BOUNDARY
.
.         ORIGINAL TREE (FOR INSTALLATION)
.
LAND      NODE      HORSE,WHALE  24  0
          'DOES IT LIVE ON THE LAND'
HORSE     NODE      0,0  7  LAND
          'A HORSE'
WHALE     NODE      0,0  7  LAND
          'A WHALE'
.
MEML      EQU       $-MEMORY
/.
.
.         DATA AREA
.
$(2).
.
SIGNON    'THINK OF AN ANIMAL.%'
SIGNL     EQU       $-SIGNON
SIGNON1   'I WILL ASK QUESTIONS AND TRY TO GUESS YOUR ANIMAL.%'
SIGNL1    EQU       $-SIGNON1
LATEM     'WE''VE GOT PROBLEMS.  PLEASE TRY AGAIN LATER.%'
LATEL     EQU       $-LATEM
EOFAM     'THANK YOU, PLEASE PLAY AGAIN SOON.%'
EOFAL     EQU       $-EOFAM
MUYM      'MAKE UP YOUR MIND.'
MUYML     EQU       $-MUYM
.
ISIT      'IS IT &'
.
P         PROC      1
QCP*      NAME      0
          *         P(1),P(1,1)
I         DO        P(1)-1 , * 0,P(1,I+1)
          END
.
WHATANI   QCP       WHT1,WHT2,WHT3,WHT4
WHT1      'WHAT WAS THE ANIMAL&'
```

```
WHT2      'WHAT ANIMAL WERE YOU THINKING OF&'
WHT3      'WHAT ANIMAL DID YOU HAVE IN MIND&'
WHT4      'WELL THEN, WHAT IS IT&'
.
YORN      QCP       YRN1,YRN2
YRN1      'YES OR NO&'
YRN2      'PLEASE ANSWER YES OR NO&'
.
PLAYAGAIN QCP       PG1,PG2
PG1       'DO YOU WANT TO TRY ANOTHER ANIMAL&'
PG2       'PLAY AGAIN'
.
BOTHWAYS  QCP       BTH1,BTH2
BTH1      'COULD YOU ANSWER EITHER WAY FOR &'
BTH2      'COULD SAY EITHER YES OR NO FOR &'
.
SPECIFY   QCP       SPC1,SPC2,SPC3
SPC1      'BE SPECIFIC.  WHAT KIND OF &'
SPC2      'PLEASE BE MORE SPECIFIC.  WHAT KIND OF &'
SPC3      'WHAT KIND OF &'
.
ASKRIGHT  QCP       AKR1,SKR2
AKR1      'NO, GIVE ME A QUESTION LIKE ''DOES IT HAVE FUR?'':&'
AKR2      'GIVE ME A YES OR NO QUESTION LINE ''DOES IT CLIMB TREES?'':&'
.
GMQ       'GIVE ME A QUESTION WHICH DISTINGUISHES & FROM &'
HOWWDYA   'HOW WOULD YOU ANSWER THAT QUESTION FOR &'
          ON        MAINTENANCE
NEWANML   QCP       NWA1
NWA1      'WHAT IS THE REPLACEMENT ANIMAL&'
RPLQUES   QCP       RPQ1
RPQ1      'REPLACEMENT QUESTION&'
DELQASM   'ASSUMED ANSWER:  YES OR NO&'
MAINTSO   'ANIMAL & TREE MAINTENANCE PROCESSOR.&'
ALSDEL    'ALSO DELETED:  &'
BADCMM    'ILLEGAL COMMAND.&'
NOTANMM   'THAT ISN''T AN ANIMAL.&'
NOTQUEM   'THAT ISN''T A QUESTION.&'
LTMTX     'ANIMAL MEMORY TREE LEVEL & LAST UPDATED ON & AT & (UPDATE #&).&'
LTMTX2    'MEMORY LENGTH:  & WORDS.  BASE NODE:  &.&'
ENDMM     'END TREE MAINTENANCE.&'
ENDML     EQU       $-ENDMM
          OFF       MAINTENANCE
.
IOP       IO$PKT,R$ 'ANIMAL$' 28,MEMORY 0
IOPW      IO$PKT,W$ 'ANIMAL$' MEML,MEMORY 0
.
ASGMEM    '@ASG,AG &'
CREMEM    '@ASG,CPV &'
CREMEM1   ',F40///10000&'
USEMEM    '@USE ANIMAL$,&'
FREEMEM   '@FREE ANIMAL$ . '
ASGAQM    '@ASG,AQGD &'
.
VOWELS    *         'A'                 TABLE OF VOWELS
          *         'E'
          *         'I'
          *         'O'
          *         'U'
VOWELL    EQU       $-VOWELS
.
.         TABLE OF SENTENCE-BEGINNING SYNTAXES WHICH MAY BE
.         TRANSFORMED INTO AN EITHER/OR QUESTION.
.
CVPT      'CAN &&'
          'DOES &'
          'WILL &'
          'MAY &&'
CVPTL     EQU       $-CVPT
.
.         TABLE OF AFFIRMATIVE-SOUNDING ANSWERS
.
YESL      'SURE'
          'OF COU'                      OF COURSE
          'OK'
          'O.K.'
          'ALL RI'                      ALL RIGHT
          'SOMETI'                      SOMETIMES (TRICKY, HUH?)
          'FREQUE'                      FREQUENTLY
          'CERTAI'                      CERTAINLY
          'SURELY'
YESLL     EQU       $-YESL
.
.         TABLE OF NEGATIVE-SOUNDING RESPONSES
.
NOL       'HARDLY'
          'HELL N'                      HELL NO
          'RARELY'
NOLL      EQU       $-NOL
.
.         TABLE OF COMMONLY-USED QUESTIONS BEGINNINGS WHICH ARE NOT
.         APPLICABLE FOR YES OR NO QUESTIONS.  IF THE USER BEGINS
.         HIS QUESTION WITH ONE OF THESE, TELL HIM WHAT KIND OF A
.         QUESTION WE'RE LOOKING FOR.
.
PREFIX    'WHO'
          'WHAT'
          'WHEN'
          'WHY'
          'HOW'
PREFXL    EQU       $-PREFIX
.
.         TABLE OF VALID ARTICLES
.
ARTT      *         'A'
          *         'AN'
          *         'THE'
          *         'LE'
          *         'LA'
          *         'UN'
          *         'UNE'
          *         'EL'
          *         'IL'
ARTTL     EQU       $-ARTT
.
.         COMMAND TABLE FOR MAINTENANCE MODE
.
          ON        MAINTENANCE
TMPCT     .                             MAINTENANCE MODE COMMAND TABLE
          *         'AB ',EOFANS        AB:  ABORT MAINTENANCE MODE
          *         'DA ',DELANML       DA:  DELETE ANIMAL
          *         'DQ ',DELQUES       DQ:  DELETE QUESTION
          *         'CA ',CHGANML       CA:  CHANGE ANIMAL
          *         'CQ ',CHGQUES       CQ:  CHANGE QUESTION
          *         'LT ',LISTREE       LT:  LIST MEMORY TREE
          *         'PL ',RESTART       PL:  PLAY A ROUND
          *         'END',RWOUT         END: TERMINATE TREE MAINTENANCE
TMPCTL    EQU       $-TMPCT
          OFF       MAINTENANCE
.
GIGO      *         0125,FL$            TREAD$ PACKET FOR STANDARD QUESTIONS
          *         EOFANS,REPLY
.
REPLPK    E$PKT     14,REPLY
.
HIGHCORE  EQUF      $,,H1               HIGHEST MEMORY ADDRESS AVAILABLE NOW
UANLW     EQUF      $,,S4               LENGTH OF USER'S ANIMAL IN WORDS
UANL      EQUF      $,,T3 (Q4)          LENGTH OF USER'S ANIMAL IN CHARACTERS
          *         LASTD$,0
.
UQLW      EQUF      $,,S1               USER QUESTION LENGTH IN WORDS
          ON        MAINTENANCE
MAINT     EQUF      $,,S2               MAINTENANCE MODE FLAG
          OFF       MAINTENANCE
NEEDRBAL  EQUF      $,,S3               TREE NEEDS REBALANCING FLAG
NEEDPACK  EQUF      $,,S4               TREE NEEDS CONDENSATION FLAG
UQL       EQUF      $,,T3 (Q4)          USER QUESTION LENGTH IN CHARACTERS
          *         0,0
.
HIGHUSE   EQUF      $,,H1               HIGHEST ADDRESS IN USE
UFIND     EQUF      $,,H2               WHERE USER'S ANIMAL WAS FOUND IN TREE
          *         0,0
.
UANML     RES       14                  USER'S ANIMAL SAVE BUFFER
USERID    RES       1                   USERID OF CALLING RUN
ACCOUNT   RES       2                   ACCOUNT OF CALLING RUN
UQUES     RES       14                  USER QUESTION BUFFER
INFOR     RES       56                  INFOR TABLE BUFFER
INFL      EQU       $-INFOR             LENGTH OF INFOR TABLE
REPLY     RES       14                  REPLY BUFFER FOR QUESTIONS
.
          END       BEGIN
```

Back to UNIVAC Memories
The ANIMAL Episode
PERVADE Source Code

by John Walker
August 13, 1996