K 1

```
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEE                    TTT        PPP         PPP
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUU         UUU  EEE                    TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP
```

```
_$


Va
---
000
000
000
7F
7F
7F
7F
7F
7F
7F
7F
7F
```

C 14

FILEID**RMSTEST8

```
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT    888888
RRRRRRRR    MM      MM    SSSSSSSS  TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT    888888
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT      88      88
RR      RR  MMMM  MMMM  SS              TT      EE          SS              TT      88      88
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT      88      88
RR      RR  MM  MM  MM  SS              TT      EE          SS              TT      88      88
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT        888888
RRRRRRRR    MM      MM    SSSSSS        TT      EEEEEEEE      SSSSSS        TT        888888
RR  RR      MM      MM          SS      TT      EE                  SS      TT      88      88
RR  RR      MM      MM          SS      TT      EE                  SS      TT      88      88
RR    RR    MM      MM          SS      TT      EE                  SS      TT      88      88      ....
RR    RR    MM      MM          SS      TT      EE                  SS      TT      88      88      ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT        888888        ....
RR      RR  MM      MM  SSSSSSSS        TT      EEEEEEEEEE  SSSSSSSS        TT        888888        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000                .EXTERNAL       REPORT_ERROR,BEG_DESCR,BEGPUT,FIN_DESCR,FINPUT,FLD_DESCR,FLD
0000                $BEGIN  RMSTEST8,002,__RMSTEST,<INDEXED FILE TEST PROGRAM>,<GBL,LONG>
0000
0000        ;
0000        ; Macros...
0000        ;
0000
0000                .MACRO  STORE STRING,PRE
0000                .SAVE
0000                .PSECT  __$RMSNAM
0000                $$.TMPX=.
0000                PRE                             ; store any carriage control info
0000                .ASCII  %STRING%
0000                $$.TMPX1=.-$$.TMPX
0000                .RESTORE
0000                .ENDM   STORE
0000
0000                .MACRO  BEGIN TSTNAM
0000                STORE   <TSTNAM>
0000                MOVL    #$$.TMPX,BEG_DESCR+4    ; addr
0000                MOVL    #$$.TMPX1,BEG_DESCR     ; len
0000                BSBW    BEGPUT
0000                .ENDM   BEGIN
0000
0000                .MACRO  FINISH TSTNAM
0000                STORE   <TSTNAM>
0000                MOVL    #$$.TMPX,FIN_DESCR+4    ; addr
0000                MOVL    #$$.TMPX1,FIN_DESCR     ; len
0000                BSBW    FINPUT
0000                .ENDM   FINISH
0000
0000                .MACRO  FIELD FLDNAM
0000                STORE   <FLDNAM>
0000                MOVL    #$$.TMPX,FLD_DESCR+4    ; addr
0000                MOVL    #$$.TMPX1,FLD_DESCR     ; len
0000                BSBW    FLDPUT
0000                .ENDM   FIELD
0000
0000                .MACRO  MBPT ?L                 ; bpt, maybe?
0000                BLBC    VERBOSITY,L
0000                BPT
0000        L:
0000                .ENDM   MBPT
0000
0000        ;
0000        ; IXVAR1 -- program to create and populate the rms-11 file
0000        ;               idxvr1.dat  ;  records are variable length
0000        ;               keys cannot change;  dupes are fixed
0000        ;
0000
0000
0000                .ENABL  DBG
0000                .NLIST  MEB
0000                $RMSDEF
0000
0000
0000        ;
```

```
                            0000   ;  buffer area
                            0000   :
                            0000
                  00000020  0000   DELBUF: .BLKB   32.                       ; store deleted record numbers
                      0038  0020   BASLEN: .WORD   56.                       ; record 0 base length (changes)
                      0001  0022   PSTEP:  .WORD   1                         ; pseudo-step buffer
                            0024
                  00000624  0024   RFABUF: .BLKW   768.                      ; rfa storage, in rec # order
                            0624
                            0624   ;
                            0624   ; Step and first have to be such that recnum is not duplicated in 255 puts
                            0624   ;
                            0624
                        FF  0624   STEP:   .BYTE   255.                      ; set byte by odt to desired step val
                        00  0625           .BYTE   0                         ; key to get 1st record on find random
                            0626
                        FF  0626   FIRST:  .BYTE   255.                      ; set byte by odt to desired step val
                            0627
                        00  0627   COUNT:  .BYTE   0
                        00  0628   LAST:   .BYTE   0
                            0629   OPERATION:
                  00000000  0629           .LONG
                            062D
                            062D   ;
                            062D   ; this is the record buffer:
                            062D   ;
                            062D
                  0000062F  062D   MASBUF: .BLKB   2                         ; rec # byte, dup control byte
4C 4B 4A 49 48 47 46 45 44 43 42 41  062F           .ASCII  /ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ/
58 57 56 55 54 53 52 51 50 4F 4E 4D  063B
4A 49 48 47 46 45 44 43 42 41 5A 59  0647
56 55 54 53 52 51 50 4F 4E 4D 4C 4B  0653
                       5A 59 58 57  065F
                            0663
                            0663   RECNUM: .REPT   131.                      ; rec # byte, nbytes fill, check byte,
                            0663           .WORD   0                         ; 5 bytes for rfa update, 255.-n bytes
                      0000  0663           .ENDM                             ; left over (n= rec#)
                            0769
                            0769   ;
                            0769   ; structure area
                            0769   ;
                            0769
                            0769
                         00000000          .PSECT  DATA,LONG
                            0000   FAB1:   $FAB    FAC=<PUT,GET,UPD,DEL>,-
                            0000                   FNM=<TST$DISK:IDXVAR1.DAT>,-
                            0000                   ORG=IDX,-
                            0000                   XAB=XAB0,-
                            0000                   ALQ=300,-
                            0000                   DEQ=100,-
                            0000                   NAM=NAMBLK,-
                            0000                   FOP=<NAM>,-
                            0000                   BKS=1
                            0050
                            0050   NAMBLK: $NAM
                            00B0
                            00B0   RAB1:   $RAB    RAC=KEY,-
```

```
00B0                           RBF=MASBUF,-
00B0                           UBF=MASBUF,-
00B0                           USZ=311,-
00B0                           KBF=STEP+1,-
00B0                           KSZ=1,-
00B0                           ROP=<KGE>,-
00B0                           MBF=10,-
00B0                           FAB=FAB1
00F4         ;
00F4         ; key is a,a-z(2),a,a,a,a,a-z(2),a-z(2),rec#
00F4         ;  therefore they are ordered by recnum
00F4         ;
00F4
00F4         XAB0:   $XABKEY NXT=XAB1,-
00F4                 POS=<2,2,2,2,2,2,2,2>,-
00F4                 SIZ=<1,52,1,1,1,1,52,53>,-
00F4                 REF=0
0140
0140         ;
0140         ; key is a,a-z(2),a,a,a,a,a-z(2),a-z(2),rec#
0140         ; therefore they are ordered by recnum
0140         ;
0140
0140         XAB1:   $XABKEY NXT=XAB2,-
0140                 POS=<2,2,2,2,2,2,2,2>,-
0140                 SIZ=<1,52,1,1,1,1,52,53>,-
0140                 REF=1
018C
018C         ;
018C         ; key is a-z(2),a-z(2),a-z(2),a,a,a,a-b,a
018C         ; therefore recnums appear in order of input
018C         ;
018C
018C         XAB2:   $XABKEY NXT=XAB3,-
018C                 POS=<2,2,2,2,2,2,2,2>,-
018C                 SIZ=<52,52,52,1,1,1,2,1>,-
018C                 REF=2
01D8
01D8         ;
01D8         ; key is a,a,a,a-z(2),a-z(2),a-z(2),a-b,(127,128)
01D8         ; therefore mostly dups, but not all
01D8         ;
01D8
01D8         XAB3:   $XABKEY NXT=XAB4,-
01D8                 POS=<2,2,2,2,2,2,2,1>,-
01D8                 SIZ=<1,1,1,52,52,52,2,1>,-
01D8                 REF=3
0224
0224         ;
0224         ; key is a-a,a-z(2),recnum
0224         ; so is ordered by recnum
0224         ;
0224
0224         XAB4:   $XABKEY NXT=XAB5,-
0224                 POS=<2,2>,-
0224                 SIZ=<27,53>,-
0224                 REF=4
```

```
                           0270
                           0270   ;
                           0270   ; key is rec# only
                           0270   ;
                           0270
                           0270   XAB5:    $XABKEY NXT=XAB6,-
                           0270            POS=<0>,-
                           0270            SIZ=<1>,-
                           0270            REF=5
                           02BC
                           02BC   ;
                           02BC   ; key is a
                           02BC   ; therefore records are in input order
                           02BC   ;
                           02BC
                           02BC   XAB6:    $XABKEY NXT=XAB7,-
                           02BC            POS=<2>,-
                           02BC            SIZ=<1>,-
                           02BC            REF=6
                           0308
                           0308   ;
                           0308   ; key is (127,128)
                           0308   ; mostly all dups
                           0308
                           0308   XAB7:    $XABKEY POS=<1>,-
                           0308            SIZ=<1>,-
                           0308            REF=7
                           0354
                           0354
                           0354   ;
                           0354   ; begin code
                           0354   ;
                           0354
                       00000769            .PSECT  __RMSTEST,LONG
                           0769
                           0769   RMT$INDEX::
                           0769
                      0000 0769   START:   .WORD   0
5A   00000000'EF     9E    076B            MOVAB   FAB1,R10                ; address of fab
                           0772            BEGIN   <INDEXED TEST>
          FE9E CF    D4    0787            CLRL    OPERATION               ; indicate create operation
                           078B            $CREATE FAB=R10,ERR=REPORT_ERROR; do create
             F863'   30    079A            BSBW    ERR
                           079D
5A   000000B0'EF     9E    079D   1$:      MOVAB   RAB1,R10
          FE81 CF    D6    07A4            INCL    OPERATION               ; indicate connect operation
                           07A8            $CONNECT RAB=R10,-
                           07A8             ERR=REPORT_ERROR               ; do connect
             F846'   30    07B7            BSBW    ERR
                           07BA
 FE6E CF    7F 8F    90    07BA   2$:      MOVB    #127.,MASBUF+1          ; set up for inserts
    53    FE62 CF    9A    07C0            MOVZBL  FIRST,R3                ; start record #
                           07C5
53    FFFFFF00 8F    CA    07C5   PNEXT:   BICL2   #^X<FFFFFF00>,R3        ; insert the records
     FE5C CF    53   90    07CC            MOVB    R3,MASBUF
     FE8D CF    53   90    07D1            MOVB    R3,RECNUM
 00000664'E3    53   90    07D6            MOVB    R3,L^RECNUM+1(R3)
```

```
                                  07DD   296           $RAB_STORE      RAB=R10,-
                                  07DD   297                           RSZ=R3
     22 AA   F83B CF        A0    07E1   298           ADDW2   BASLEN,RAB$W_RSZ(R10)
        FE3D CF     02      9A    07E7   299           MOVZBL  #2,OPERATION              ; indicate put
                                  07EC   300           $PUT    RAB=R10,ERR=REPORT_ERROR
                  F802'     30    07FB   301           BSBW    ERR
              50     06     9A    07FE   302           MOVZBL  #6,R0
              50     53     C4    0801   303           MULL2   R3,R0
                  10 AA     D0    0804   304           MOVL    RAB$L_RFA0(R10),-
            00000024'E0           0807   305                    L^RFABUF(R0)              ; move rfa of record
                  14 AA     B0    080C   306           MOVW    RAB$W_RFA4(R10),-
            00000028'E0           080F   307                    L^RFABUF+4(R0)            ; 6 char long
   FE13 CF   FE16 CF        92    0814   308           MCOMB   MASBUF+1,MASBUF+1
            00000664'E3     94    081B   309           CLRB    L^RECNUM+1(R3)
        50   FDFF CF        9A    0821   310           MOVZBL  STEP,R0
              53     50     C0    0826   311           ADDL2   R0,R3
        FDF9 CF     04      80    0829   312           ADDB2   #4,COUNT
                     95     12    082E   313           BNEQ    PNEXT
   FDF2 CF    53     50     83    0830   314           SUBB3   R0,R3,LAST                ; move last key put into  last
            00000A59'EF     16    0836   315           JSB     CHKALL                    ;  check all the indexes
                                  083C   316
                                  083C   317           $RAB_STORE      RAB=R10,-
                                  083C   318                           RAC=KEY
              FDE5 CF       D6    0840   319           INCL    OPERATION
                                  0844   320           $FIND   RAB=R10,ERR=REPORT_ERROR
                  F7AA'     30    0853   321           BSBW    ERR
                                  0856   322           $RAB_STORE      RAB=R10,-
                                  0856   323                           RAC=SEQ
        53    FDCA CF       9A    085A   324           MOVZBL  LAST,R3
        54    53     06     C5    085F   325           MULL3   #6,R3,R4
   54   00000024'8F         C0    0863   326           ADDL2   #RFABUF,R4
        F7B1 CF     3D      B0    086A   327           MOVW    #61.,BASLEN
                                  086F   328
        FDB5 CF     04      9A    086F   329 UNEXT:    MOVZBL  #4,OPERATION
                                  0874   330           $GET    RAB=R10,ERR=REPORT_ERROR
                  F77A'     30    0883   331           BSBW    ERR
        56    FDA3 CF       9A    0886   332           MOVZBL  MASBUF,R6
              53     56     D1    088B   333           CMPL    R6,R3
                     15     13    088E   334           BEQL    DAT_OK
                                  0890   335           FIELD   <DATA>
                                  08A5   336
        50    FDBC CF       DE    08A5   337 DAT_OK:   MOVAL   RECNUM+2,R0
              50     53     C0    08AA   338           ADDL2   R3,R0                     ; position to end of buffer
              80     84     D0    08AD   339           MOVL    (R4)+,(R0)+               ; move rfa in for update
              80     84     90    08B0   340           MOVB    (R4)+,(R0)+
                     54     D6    08B3   341           INCL    R4
           22 AA     05     A0    08B5   342           ADDW2   #5,RAB$W_RSZ(R10)
        FD6B CF      05     9A    08B9   343           MOVZBL  #5,OPERATION
                                  08BE   344           $UPDATE RAB=R10,ERR=REPORT_ERROR
                  F730'     30    08CD   345           BSBW    ERR
        50    FD91 CF       DE    08D0   346           MOVAL   RECNUM+2,R0               ; position to end of buffer
              50     53     C0    08D5   347           ADDL2   R3,R0
                  FF A0     94    08D8   348           CLRB    -1(R0)
          60   10 AA        D1    08DB   349           CMPL    RAB$L_RFA0(R10),(R0)      ; check rfa after update
                     15     13    08DF   350           BEQL    RFA_OK
                                  08E1   351           FIELD   <DATA (RFA) AFTER UPDATE>
                                  08F6   352
```

```
                      80   D4  08F6   353 RFA_OK: CLRL    (R0)+
           60     14 AA    91  08F8   354         CMPB    RAB$W_RFA4(R10),(R0)
                      15   13  08FC   355         BEQL    RFA_OK1
                               08FE   356         FIELD   <DATA (RFA) AFTER UPDATE>
                               0913   357 RFA_OK1:
                      60   94  0913   358         CLRB    (R0)
                      53   96  0915   359         INCB    R3                         ; r3 started as last key put
                               0917   360                                            ; and they increase by 1
                      03   13  0917   361         BEQL    10$
                    FF53   31  0919   362         BRW     UNEXT
                               091C   363 10$:    $RAB_STORE      RAB=R10,-
                               091C   364                         USZ=#316.
         F6FB CF      01   B0  0922   365         MOVW    #1,PSTEP
          00000A59'EF      16  0927   366         JSB     CHKALL
                               092D   367
                               092D   368 ;
                               092D   369 ; test delete
                               092D   370 ;
                               092D   371
              FCF6 CF      94  092D   372         CLRB    COUNT
           53 FCF3 CF      9A  0931   373         MOVZBL  LAST,R3
                               0936   374         $RAB_STORE      RAB=R10,RAC=RFA
                               093A   375
         FCEA CF      04   9A  093A   376 DNEXT:  MOVZBL  #4,OPERATION
              50      06   9A  093F   377         MOVZBL  #6,R0
              50      53   C4  0942   378         MULL2   R3,R0
          00000024'E0      D0  0945   379         MOVL    L^RFABUF(R0),-
                   10 AA       094B   380                  RAB$L_RFA0(R10)          ; move rfa into rab for get
  14 AA   00000028'E0      B0  094D   381         MOVW    L^RFABUF+4(R0),RAB$W_RFA4(R10)
                               0955   382         $GET    RAB=R10,ERR=REPORT_ERROR; get the record, locking it
                  F699'    30  0964   383         BSBW    ERR
           50 FCBC CF      9A  0967   384         MOVZBL  COUNT,R0
 00000000'E0  FCBD CF      90  096C   385         MOVB    MASBUF,L^DELBUF(R0)      ; store away its rec number
         FCAF CF      06   9A  0975   386         MOVZBL  #6,OPERATION
                               097A   387         $DELETE RAB=R10,ERR=REPORT_ERROR
                  F674'    30  0989   388         BSBW    ERR
              FC97 CF      96  098C   389         INCB    COUNT
              53      08   80  0990   390         ADDB2   #8,R3
                      A5   12  0993   391         BNEQ    DNEXT                      ; deleted 32 records (every 8th)
                               0995   392
                               0995   393 ;
                               0995   394 ; check deleted records no longer appear anywhere
                               0995   395 ;
                               0995   396
           35 AA      07   90  0995   397         MOVB    #7,RAB$B_KRF(R10)          ; do all 8 keys
                               0999   398         $RAB_STORE      RAB=R10,RAC=SEQ
                               099D   399
         FC87 CF      08   9A  099D   400 KNEXT:  MOVZBL  #8.,OPERATION
                               09A2   401         $REWIND RAB=R10,ERR=REPORT_ERROR; rewind on new key of reference
                  F64C'    30  09B1   402         BSBW    ERR
              53      38   D0  09B4   403         MOVL    #56.,R3                    ; get all the records sequentially
         FC6D CF      04   9A  09B7   404         MOVZBL  #4,OPERATION
                               09BC   405
                               09BC   406 GNEXTD: $GET    RAB=R10,ERR=REPORT_ERROR
                  F632'    30  09CB   407         BSBW    ERR
         56   F62E CF      DE  09CE   408         MOVAL   DELBUF,R6
         50   56      08   C1  09D3   409         ADDL3   #8.,R6,R0
```

```
                                    09D7     410 CHK_DEL:
        FC51 CF    86     91        09D7     411          CMPB     (R6)+,MASBUF            ; make sure it wasn't deleted
                   15     12        09DC     412          BNEQ     DEL_OK
                                    09DE     413          FIELD    <DATA (RECORD WAS DELETED)>
            50     56     D1        09F3     414 DEL_OK:  CMPL     R6,R0
                   DF     1F        09F6     415          BLSSU    CHK_DEL
                C1 53     F5        09F8     416          SOBGTR   R3,GNEXTD
                                    09FB     417          $GET     RAB=R10                 ; verify end of file
   0001827A 8F     50     D1        0A04     418          CMPL     R0,#RMS$_EOF
                   0B     13        0A0B     419          BEQL     EOF
               F5F0'      30        0A0D     420          BSBW     EOFPUT
                                    0A10     421          MBPT
                                    0A18     422
                                    0A18     423 EOF:
               35 AA      97        0A18     424          DECB     RAB$B_KRF(R10)
                   80     18        0A1B     425          BGEQ     KNEXT
        FC07 CF    07     9A        0A1D     426          MOVZBL   #7,OPERATION            ; indicate close
     5A   00000000'EF     DE        0A22     427          MOVAL    FAB1,R10
  04 AA   01008000 8F     D0        0A29     428          MOVL     #<FAB$M_NAM!FAB$M_DLT>,FAB$L_FOP(R10)
                                    0A31     429          $CLOSE   FAB=R10,ERR=REPORT_ERROR
               F5BD'      30        0A40     430          BSBW     ERR
                                    0A43     431          FINISH   <INDEXED TEST>
                          04        0A58     432          RET
                                    0A59     433
                                    0A59     434
                                    0A59     435 ;
                                    0A59     436 ; subroutines
                                    0A59     437 ;
                                    0A59     438
                                    0A59     439 ;
                                    0A59     440 ; this routine checks the indices in various ways
                                    0A59     441 ;
                                    0A59     442
       53    FBCB CF      9A        0A59     443 CHKALL:  MOVZBL   LAST,R3                 ; set up to check index on key 0
                0093      30        0A5E     444          BSBW     CHKIDX                  ; and do so
       53    FBC3 CF      9A        0A61     445          MOVZBL   LAST,R3                 ; set up to check index on key 1
               35 AA      96        0A66     446          INCB     RAB$B_KRF(R10)
                0088      30        0A69     447          BSBW     CHKIDX
       53    FBB8 CF      9A        0A6C     448          MOVZBL   LAST,R3                 ; set up to check index on key 4
                                    0A71     449          $RAB_STORE       RAB=R10,-
                                    0A71     450                           KRF=#4
                007C      30        0A75     451          BSBW     CHKIDX                  ; and do so
       53    FBAC CF      9A        0A78     452          MOVZBL   LAST,R3                 ; set up to check index on key 5
               35 AA      96        0A7D     453          INCB     RAB$B_KRF(R10)
                0071      30        0A80     454          BSBW     CHKIDX                  ; and do so
  F598 CF    FB9D CF      9B        0A83     455          MOVZBW   STEP,PSTEP              ; set up to check index on key 6
       53    FB98 CF      9A        0A8A     456          MOVZBL   FIRST,R3
               35 AA      96        0A8F     457          INCB     RAB$B_KRF(R10)
                005F      30        0A92     458          BSBW     CHKIDX                  ; and do so
       53    FB8D CF      9A        0A95     459          MOVZBL   FIRST,R3                ; set up to check index on key 2
                                    0A9A     460          $RAB_STORE       RAB=R10,-
                                    0A9A     461                           KRF=#2
                  54      10        0A9E     462          BSBB     CHKIDX                  ; and do so
          50      02      3C        0AA0     463          MOVZWL   #2,R0
       F57A CF    50      A4        0AA3     464          MULW2    R0,PSTEP
       53    FB7A CF      9A        0AA8     465          MOVZBL   FIRST,R3                ; start at beginning
               35 AA      96        0AAD     466          INCB     RAB$B_KRF(R10)          ; index =3
        FB71 CF    80 8F    90 0AB0    467          MOVB    #128.,COUNT               ; check half
                      3C    10 0AB6    468          BSBB    CHKIDX
         53    FB6A CF      9A 0AB8    469          MOVZBL  FIRST,R3                  ; setup to check second half
         53    FB63 CF      80 0ABD    470          ADDB2   STEP,R3
        FB5F CF    80 8F    90 0AC2    471          MOVB    #128.,COUNT
                      46    10 0AC8    472          BSBB    GNEXT                     ; cont where left off
         53    FB58 CF      9A 0ACA    473          MOVZBL  FIRST,R3
                               0ACF    474          $RAB_STORE      RAB=R10,-
                               0ACF    475                          KRF=#7
        FB4E CF    80 8F    90 0AD3    476          MOVB    #128.,COUNT
                      19    10 0AD9    477          BSBB    CHKIDX
         53    FB47 CF      9A 0ADB    478          MOVZBL  FIRST,R3
         56    FB40 CF      9A 0AE0    479          MOVZBL  STEP,R6
               53    56     C0 0AE5    480          ADDL2   R6,R3
        FB39 CF    80 8F    90 0AE8    481          MOVB    #128.,COUNT
                      20    10 0AEE    482          BSBB    GNEXT
                   35 AA    94 0AF0    483          CLRB    RAB$B_KRF(R10)
                            05 0AF3    484          RSB
                               0AF4    485
                               0AF4    486  ;
                               0AF4    487  ; this routine checks one index
                               0AF4    488  ; input:         r3 - start record number
                               0AF4    489  ;
                               0AF4    490
                               0AF4    491  CHKIDX: $RAB_STORE      RAB=R10,-
                               0AF4    492                          RAC=KEY
         FB2C CF    03      9A 0AF8    493          MOVZBL  #3,OPERATION              ; indicate find function
                               0AFD    494          $FIND   RAB=R10,ERR=REPORT_ERROR
                               0B0C    495          $RAB_STORE      RAB=R10,-
                               0B0C    496                          RAC=SEQ
                               0B10    497
      53    FFFFFF00 8F     CA 0B10    498  GNEXT:  BICL2   #^X<FFFFFF00>,R3
         FB0D CF    04      9A 0B17    499          MOVZBL  #4,OPERATION              ; indicate get function
                               0B1C    500          $GET    RAB=R10,ERR=REPORT_ERROR
         56    FAFE CF      9A 0B2B    501          MOVZBL  MASBUF,R6
               53    56     91 0B30    502          CMPB    R6,R3
                     15     13 0B33    503          BEQL    D_OK
                               0B35    504          FIELD   <DATA>
                               0B4A    505
      53    00000664'E3     91 0B4A    506  D_OK:   CMPB    L^RECNUM+1(R3),R3
                     15     13 0B51    507          BEQL    D_OK1
                               0B53    508          FIELD   <DATA>
 50   22 AA    F4B4 CF      A3 0B68    509  D_OK1:  SUBW3   BASLEN,RAB$W_RSZ(R10),R0; calc rec # from record length
               53    50     B1 0B6F    510          CMPW    R0,R3
                     15     13 0B72    511          BEQL    RSZ
                               0B74    512          FIELD   <RSZ>
                               0B89    513
               50    06     9A 0B89    514  RSZ:    MOVZBL  #6,R0
               50    53     C4 0B8C    515          MULL2   R3,R0
  00000024'E0    10 AA      D1 0B8F    516          CMPL    RAB$L_RFA0(R10),L^RFABUF(R0)
                     15     13 0B97    517          BEQL    RFA_OK100
                               0B99    518          FIELD   <RFA>
                               0BAE    519
                               0BAE    520  RFA_OK100:
  00000028'E0    14 AA      B1 0BAE    521          CMPW    RAB$W_RFA4(R10),L^RFABUF+4(R0)
                     15     13 0BB6    522          BEQL    RFA_OK101
                               0BB8    523          FIELD   <RFA>
                            0BCD   524 RFA_OK101:
56    F451 CF    3C         0BCD   525         MOVZWL  PSTEP,R6
        53   56  C0         0BD2   526         ADDL2   R6,R3
FA4D CF      04  80         0BD5   527         ADDB2   #4,COUNT
             03  13         0BDA   528         BEQL    10$
           FF31  31         0BDC   529         BRW     GNEXT
                 05         0BDF   530 10$:    RSB
                            0BE0   531
                            0BE0   532 ;
                            0BE0   533 ; this is the error subroutine
                            0BE0   534 ; operation      -       0 create
                            0BE0   535 ;                        1 connect
                            0BE0   536 ;                        2 put
                            0BE0   537 ;                        3 find
                            0BE0   538 ;                        4 get
                            0BE0   539 ;                        5 update
                            0BE0   540 ;                        6 delete
                            0BE0   541 ;                        7 close
                            0BE0   542 ;                        8 rewind
                            0BE0   543 ;
                            0BE0   544
                            0BE0   545         .END
```

```
$$.DTPTMP                      = 00000000    D           FIN_DESCR                        ********    X   00
$$.PSECT_EP                    = 00000000                FIRST                            00000626 R  D   01
$$.SPTMP                       = 00000000    D           FLDPUT                           ********    X   00
$$.STMP                        = 00000001    D           FLD_DESCR                        ********    X   00
$$.TAB                         = 00000308 R  D   03      GNEXT                            00000B10 R  D   01
$$.TABEND                      = 00000354 R  D   03      GNEXTD                           000009BC R  D   01
$$.TMP                         = 00000000    D           KNEXT                            0000099D R  D   01
$$.TMP1                        = 00000002    D           LAST                             00000628 R  D   01
$$.TMP2                        = 0000005A                MASBUF                           0000062D R  D   01
$$.TMPX                        = 00000071 R  D   05      NAM$B_ESS                      = 0000000A    D
$$.TMPX1                       = 00000003    D           NAM$B_NOP                      = 00000008    D
$$RMSTEST                      = 0000001E                NAM$B_RSS                      = 00000002    D
$$RMS_PBUGCHK                  = 00000010                NAM$C_BID                      = 00000002    D
$$RMS_TBUGCHK                  = 00000008                NAM$C_BLN                      = 00000060    D
$$RMS_UMODE                    = 00000004                NAM$L_ESA                      = 0000000C    D
..AFLG                         = 00000000    D           NAM$L_RSA                      = 00000004    D
..FLG                          = 00000002    D           NAMBLK                           00000050 R  D   03
..MOD                          = 00000001    D           OPERATION                        00000629 R  D   01
..TYP                          = 00000003                PNEXT                            000007C5 R  D   01
.LEN                           = 00000001    D           PSTEP                            00000022 R  D   01
PASLEN                           00000020 R  D   01      RAB$B_KRF                      = 00000035    D
BEGPUT                           ********    X   00      RAB$B_RAC                      = 0000001E    D
BEG_DESCR                        ********    X   00      RAB$C_BID                      = 00000001    D
CHKALL                           00000A59 R  D   01      RAB$C_BLN                      = 00000044    D
CHKIDX                           00000AF4 R  D   01      RAB$C_KEY                      = 00000001    D
CHK_DEL                          000009D7 R  D   01      RAB$C_RFA                      = 00000002    D
COUNT                            00000627 R  D   01      RAB$C_SEQ                      = 00000000    D
DAT_OK                           000008A5 R  D   01      RAB$L_CTX                      = 00000018    D
DELBUF                           000000C) R  D   01      RAB$L_RFA0                     = 00000010    D
DEL_OK                           000009F3 R  C   01      RAB$L_ROP                      = 00000004    D
DNEXT                            0000093A R  D   01      RAB$V_KGE                      = 00000015    D
D_OK                             00000B4A R  D   01      RAB$W_RFA4                     = 00000014    D
D_OK1                            00000B68 R  D   01      RAB$W_RSZ                      = 00000022    D
EOF                              00000A18 R  D   01      RAB$W_USZ                      = 00000020    D
EOFPUT                           ********    X   00      RAB1                             000000B0 R  D   03
ERR                              ********    X   00      RECNUM                           00000663 R  D   01
FAB$B_FNS                      = 00000034    D           REPORT_ERROR                     ********    X   00
FAB$C_BID                      = 00000003    D           RFABUF                           00000024 R  D   01
FAB$C_BLN                      = 00000050    D           RFA_OK                           000008F6 R  D   01
FAB$C_IDX                      = 00000020    D           RFA_OK1                          00000913 R  D   01
FAB$C_VAR                      = 00000002    D           RFA_OK100                        00000BAE R  D   01
FAB$L_ALQ                      = 00000010    D           RFA_OK101                        00000BCD R  D   01
FAB$L_FNA                      = 0000002C    D           RMS$_EOF                       = 0001827A    D
FAB$L_FOP                      = 00000004    D           RMT$INDEX                        00000769 RG D   01
FAB$M_DLT                      = 00008000    D           RSZ                              00000B89 R  D   01
FAB$M_NAM                      = 01000000    D           START                            00000769 R  D   01
FAB$V_CHAN_MODE                = 00000002    D           STEP                             00000624 R  D   01
FAB$V_DEL                      = 00000002    D           SYS$CLOSE                        ********   GX   01
FAB$V_FILE_MODE                = 00000004    D           SYS$CONNECT                      ********   GX   01
FAB$V_GET                      = 00000001    D           SYS$CREATE                       ********   GX   01
FAB$V_LNM_MODE                 = 00000000    D           SYS$DELETE                       ********   GX   01
FAB$V_NAM                      = 00000018    D           SYS$FIND                         ********   GX   01
FAB$V_PUT                      = 00000000    D           SYS$GET                          ********   GX   01
FAB$V_UPD                      = 00000003    D           SYS$PUT                          ********   GX   01
FAB$W_GBC                      = 00000048    D           SYS$REWIND                       ********   GX   01
FAB1                             00000000 R  D   03      SYS$UPDATE                       ********   GX   01
FINPUT                           ********    X   00      UNEXT                            0000086F R  D   01
```

```
VERBOSITY                         ********    X    00
XAB$B_DAN                       = 0000000A     D
XAB$B_DTP                       = 00000013     D
XAB$B_FLG                       = 00000012     D
XAB$B_IAN                       = 00000008     D
XAB$B_LAN                       = 00000009     D
XAB$B_NUL                       = 00000015     D
XAB$B_PROLOG                    = 00000048     D
XAB$B_REF                       = 00000017     D
XAB$B_SIZ0                      = 0000002E     D
XAB$B_SIZ1                      = 0000002F     D
XAB$B_SIZ2                      = 00000030     D
XAB$B_SIZ3                      = 00000031     D
XAB$B_SIZ4                      = 00000032     D
XAB$B_SIZ5                      = 00000033     D
XAB$B_SIZ6                      = 00000034     D
XAB$B_SIZ7                      = 00000035     D
XAB$C_KEY                       = 00000015     D
XAB$C_KEYLEN                    = 0000004C     D
XAB$C_STG                       = 00000000     D
XAB$L_KNM                       = 00000038     D
XAB$L_NXT                       = 00000004     D
XAB$V_CHG                       = 00000001     D
XAB$V_DUP                       = 00000000     D
XAB$W_DFL                       = 0000001C     D
XAB$W_IFL                       = 0000001A     D
XAB$W_POS0                      = 0000001E     D
XAB$W_POS1                      = 00000020     D
XAB$W_POS2                      = 00000022     D
XAB$W_POS3                      = 00000024     D
XAB$W_POS4                      = 00000026     D
XAB$W_POS5                      = 00000028     D
XAB$W_POS6                      = 0000002A     D
XAB$W_POS7                      = 0000002C     D
XAB0                              000000F4 R   D   03
XAB1                              00000140 R   D   03
XAB2                              0000018C R   D   03
XAB3                              000001D8 R   D   03
XAB4                              00000224 R   D   03
XAB5                              00000270 R   D   03
XAB6                              000002BC R   D   03
XAB7                              00000308 R   D   03

                                                   +----------------+
                                                   ! Psect synopsis !
                                                   +----------------+

PSECT name                        Allocation            PSECT No.  Attributes
----------                        ----------            ---------  ----------
.  ABS  .                         00000000  (     0.)   00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
__RMSTEST                         00000BE0  (  3040.)   01 (  1.)  NOPIC   USR   CON   REL   GBL NOSHR   EXE   RD    WRT NOVEC LONG
$ABS$                             00000000  (     0.)   02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
DATA                              00000354  (   852.)   03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG
$RMSNAM                           00000014  (    20.)   04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
__$RMSNAM                         00000074  (   116.)   05 (  5.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
```

```
                                            +-------------------------+
                                            ! Performance indicators !
                                            +-------------------------+

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       - ----------
Initialization                   29    00:00:00.08    00:00:01.10
Command processing              105    00:00:00.52    00:00:02.05
Pass 1                          403    00:00:19.93    00:00:53.16
Symbol table sort                 8    00:00:01.03    00:00:02.17
Pass 2                          124    00:00:03.96    00:00:09.57
Symbol table output              19    00:00:00.15    00:00:00.56
Psect synopsis output             3    00:00:00.03    00:00:00.03
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            693    00:00:25.71    00:01:08.65

The working set limit was 1350 pages.
95037 bytes (186 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 861 non-local and 5 local symbols.
545 source lines were read in Pass 1, producing 67 object records in Pass 2.
58 pages of virtual memory were used to define 44 macros.

                                           +----------------------------+
                                           ! Macro library statistics !
                                           +----------------------------+

Macro library name                            Macros defined
------------------                            --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                            0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                        34
TOTALS (all libraries)                                   34

1333 GETS were required to define 34 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:RMSTEST8/OBJ=OBJ$:RMSTEST8 MSRC$:RMSTEST8/UPDATE=(ENH$:RMSTEST8)+EXECML$/LIB
```

0409 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RMSTEST4
LIS

RMSTEST7
LIS

RMSTEST8
LIS

RMSTESTR
LIS

RMSTEST0
LIS

RMSTEST2
LIS

RMSTEST3
LIS

RMSTEST5
LIS

RMSTEST6
LIS

RMSTESTT
LIS

RMSTEST1
LIS

SATSSF17
LIS