```
 5708  006703' 001000  000345          17200          PUSH    H               ;SAVE THIS #
 5709  006704' 001000  000011          17220          DAD     B               ;ADD ON THE AMOUNT OF SPACE
 5710                                  17240                                  ;EXTRA NOW BEING USED
 5711  006705' 001000  000301          17260          POP     B               ;POP OFF HIGH ADDRESS TO MOVE
 5712  006706' 001000  000345          17280          PUSH    H               ;SAVE NEW CANDIDATE FOR STREND
 5713  006707' 001000  000315          17300          CALL    BLTU            ;BLOCK TRANSFER AND MAKE SURE
 5714  006710' 000000  002005'
 5715  006711' 000000  006701'
 5716                                  17320                                  ;WE ARE NOT OVERFLOWING THE
 5717                                  17340                                  ;STACK SPACE
 5718  006712' 001000  000341          17360          POP     H               ;[H,L]=NEW STREND
 5719  006713' 001000  00004E          17380          SHLD    STREND          ;STORE SINCE WAS OK
 5720  006714' 000000  001625'
 5721  006715' 000000  006710'
 5722                                  17400                                  ;THERE WAS ROOM, AND BLOCK TRANSFER
 5723                                  17420                                  ;WAS DONE, SO UPDATE POINTERS
 5724  006716' 001000  000140          17440          MOV     H,B             ;GET BACK [H,L] POINTING AT THE END
 5725  006717' 001000  000151          17460          MOV     L,C             ;OF THE NEW VARIABLE
 5726  006720' 001000  00004E          17480          SHLD    ARYTAB          ;UPDATE THE ARRAY TABLE POINTER
 5727  006721' 000000  001623'
 5728  006722' 000000  006714'
 5729  006723' 001000  000053          17500  ZEROEK: DCX     H               ;[H,L] IS RETURNED POINTING TO THE
 5730  006724' 001000  000066          17520          MVI     M,0             ;END OF THE VARIABLE SO WE
 5731  006725' 000000  000000
 5732  006726' 001000  000347          17540          COMPAR                  ;ZERO BACKWARDS TO [D,E] WHICH
 5733  006727' 001000  000302          17560          JNZ     ZEROEK          ;POINTS TO THE START OF THE VARIABLE
 5734  006730' 000000  006723'
 5735  006731' 000000  006721'
 5736                                  17580  IFE     LENGTH-2,<
 5737  006732' 001000  000321          17600          POP     D               ;[E]=VALTYP
 5738  006733' 001000  000163          17620          MOV     M,E             ;STORE AS PART OF THE LOOKS
 5739  006734' 001000  000043          17640          INX     H>
 5740  006735' 001000  000321          17660          POP     D
 5741  006736' 001000  000163          17680          MOV     M,E             ;PUT DESCRIPTION
 5742  006737' 001000  000043          17700          INX     H
 5743  006740' 001000  000162          17720          MOV     M,D             ;OF THIS VARIABLE
 5744                                  17740                                  ;INTO MEMORY
 5745                                  17760  IFE     LENGTH-2,<
 5746  006741' 001000  000353          17780          XCHG                    ;POINTER AT VARIABLE INTO [D,E]
 5747  006742' 001000  000023          17800  FINPTR: INX     D>              ;POINT AT THE VALUE
 5748                                  17820  IFN     LENGTH-2,<
 5749                                  17840          INX     H
 5750                                  17860  FINPTR: XCHG>                   ;VARIABLE POINTER INTO [D,E]
 5751  006743' 001000  000341          17880          POP     H               ;RESTORE THE TEXT POINTER
 5752  006744' 001000  000311          17900          RET
 5753
 5754                                  17940  IFE     MULDIM,<
 5755                                  17960  ISARY:  PUSH    B               ;REMEMBER WHAT VARIABLE LOOKS
 5756                                  17980                                  ;LIKE
 5757                                  18000  IFN     STRING,<
 5758                                  18020          PUSH    H               ;SAVE THE TXTPTR
 5759                                  18040          LHLD    DIMFLG          ;[L]=DIMFLG [H]=VALTYP
 5760                                  18060          XTHL>                   ;PUT VALTYP AND DIMFLG ON THE STACK
```



```
 5761                                  18080                                  ;AND RESTORE THE TEXT POINTER
 5762                                  18100  IFE     STRING,<
 5763                                  18120          LDA     DIMFLG          ;SINCE THIS CALL IS RECURSIVE
 5764                                  18140          PUSH    PSW>            ;DIMFLG MUST BE SAVED ON THE STACK
 5765                                  18160          CALL    INTIDX          ;EVALUATE THE INDEX INTO [D,E]
 5766                                  18180          SYNCHK  ")"             ;MAKE SURE HE CLOSED IT
 5767                                  18200  IFN     STRING,<
 5768                                  18220          XTHL                    ;[L]=DIMFLG [H]=VALTYP
 5769                                  18240                                  ;TEXT POINTER ONTO THE STACK
 5770                                  18260          SHLD    DIMFLG          ;SAVE BOTH VALUES BACK
 5771                                  18280          POP     H>              ;RESTORE THE TEXT POINTER
 5772                                  18300  IFE     STRING,<
 5773                                  18320          POP     PSW             ;GET DIMFLG OFF THE STACK
 5774                                  18340          STA     DIMFLG>         ;RESTORE IT
 5775                                  18360          XTHL                    ;[H,L] GET VARIABLE DESCRIPTOR
 5776                                  18380                                  ;TEXT POINTER IS PUT ONTO
 5777                                  18400                                  ;THE STACK
 5778                                  18420          XCHG                    ;[D,E]=DESCRIPTOR
 5779                                  18440                                  ;[H,L]=INDEX
 5780                                  18460          DAD     H               ;MULTIPLY BY 4 TO GET
 5781                                  18480          DAD     H               ;BYTE OFFSET
 5782                                  18500          PUSH    H               ;SAVE THE INDEX
 5783                                  18520          LHLD    ARYTAB          ;PLACE TO START SEARCH
 5784                                  18540          XWD     "D1000,1        ;"LXI B," OVER THE NEXT 2
 5785                                  18560  LOPFD2: POP     B               ;[B,C]=LENGTH OF LAST VARIABLE
 5786                                  18580          DAD     B               ;SKIP OVER LAST VARIABLE BY ADDING ITS
 5787                                  18600                                  ;LENGTH ONTO [H,L]
 5788                                  18620          XCHG                    ;[D,E] GET CURRENT SEARCH POINT
 5789                                  18640          PUSH    H               ;SAVE THE VARIABLE LOOK
 5790                                  18660          LHLD    STREND          ;GET PLACE TO STOP
 5791                                  18680          COMPAR                  ;SEE IF WE ARE THERE
 5792                                  18700          XCHG                    ;[H,L] GETS SEARCH POINT
 5793                                  18720          POP     D               ;POP OFF VARIABLE LOOKS
 5794                                  18740          JZ      NOTFDD          ;COULDN'T FIND IT
 5795                                  18760          PUSHM                   ;PUT ON LOOKS OF VARIABLE
 5796                                  18780                                  ;WE ARE EXAMINING
 5797                                  18800          XTHL                    ;PUT [H,L] ON THE STACK AND
 5798                                  18820                                  ;LOOKS OF VARIABLE WE ARE
 5799                                  18840                                  ;EXAMINING INTO [H,L]
 5800                                  18860          COMPAR                  ;IS THIS THE VARIABLE
 5801                                  18880          POP     H               ;POP OFF SEARCH POINTER
 5802                                  18900          PUSHM                   ;PUSH LENGTH OF VARIABLE
 5803                                  18920                                  ;BEING EXAMINED ONTO THE STACK
 5804                                  18940          JNZ     LOPFD2          ;IF NO MATCH,GO LOOK SOMEMORE
 5805                                  18960          LDA     DIMFLG          ;IS THIS VARIABLE TRYING TO BE
 5806                                  18980                                  ;DIMENSIONED AND ALREADY
 5807                                  19000          ORA     A               ;EXISTS?
 5808                                  19020          MVI     E,ERRDD         ;THATS ERROR ERRDD
 5809                                  19040          JNZ     ERROR
 5810                                  19060  MAKOFN: POP     D               ;POP OFF LENGTH OF THIS VARIABLE
 5811                                  19080          DCX     D               ;DECREMENT LENGTH SO WE CAN
 5812                                  19100                                  ;JUST LOOK AT 'CARRY' AFTER
 5813                                  19120                                  ;CALLING COMPAR
```

```
                                        19140           XTHL                    ;TRADE POINTER AT VARIABLE WITH
                                        19160                                   ;INDEX INTO THE VARIABLE
                                        19180           COMPAR                  ;SEE IF INDEX IS TOO BIG
                                        19200           MVI     E,ERRBS         ;THATS ERROR ERRBS
                                        19220           JNC     ERROR           ;SINCE LENGTH REALLY HAS AN
                                        19240                                   ;EXTRA ONE ADDED TO IT
                                        19260                                   ;IF INDEX=LENGTH DOESN'T CARRY
                                        19280                                   ;HE IS IN TROUBLE
                                        19300           POP     D               ;POP OFF POINTER AT VARIABLE
                                        19320           DAD     D               ;ADD IT TO THE INDEX
                                        19340           POP     D               ;POP OFF TEXT POINTER
                                        19360           XCHG                    ;TEXT POINTER INTO (H,L)
                                        19380           RET                     ;VARIABLE POINTER INTO (D,E)
                                        19400
                                        19420   NOTFDD: MOV     M,E             ;PUT LOOKS DOWN
                                        19440           INX     H
                                        19460           MOV     M,D
                                        19480           INX     H
                                        19500           LXI     D,$CODE+44      ;DEFAULT SIZE IS 10
                                        19520           LDA     DIMFLG          ;ARE WE DIMENSIONING
                                        19540           ORA     A
                                        19560           JZ      NOTDIM
                                        19580           POP     D               ;POP OFF INDEX
                                        19600           PUSH    D               ;PUT INDEX BACK ON
                                        19620           INX     D
                                        19640           INX     D
                                        19660           INX     D
                                        19680           INX     D
                                        19700   NOTDIM: PUSH    D
                                        19720           MOV     M,E             ;PUT LENGTH DOWN
                                        19740           INX     H
                                        19760           MOV     M,D
                                        19780           INX     H
                                        19800           PUSH    H
                                        19820           DAD     D
                                        19840           CALL    REASON          ;MAKE SURE WE'RE NOT RUNNING
                                        19860                                   ;INTO THE STACK
                                        19880           SHLD    STREND          ;SETUP NEW STORAGE END
                                        19900           POP     D
                                        19920   ZERIT2: DCX     H
                                        19940           MVI     M,0
                                        19960           COMPAR
                                        19980           JNZ     ZERIT2
                                        20000           JMP     MAKOFN>         ;FINISH UP
                                        20020
                                        20040   PAGE
```



```
                                        20060   SUBTTL  MULTIPLE DIMENSION CODE
                                        20080
                                        20100   IFN     MULDIM,<
                                        20120   ;
                                        20140   ; FORMAT OF ARRAYS IN CORE
                                        20160   ;
                                        20180   ; DESCRIPTOR
                                        20200   ;       LOW BYTE = SECOND CHARCTER (200 BIT IS STRING FLAG)
                                        20220   ;       HIGH BYTE = FIRST CHARACTER
                                        20240   ; LENGTH OF ARRAY IN CORE IN BYTES (DOES NOT INCLUDE DESCRIPTOR)
                                        20260   ; NUMBER OF DIMENSIONS 1 BYTE
                                        20280   ; FOR EACH DIMENSION STARTING WITH THE FIRST A LIST
                                        20300   ; (2 BYTES EACH) OF THE MAX INDICE+1
                                        20320   ; THE VALUES
                                        20340   ;
006745' 001000  000345  20360   ISARY:  PUSH    H               ;SAVE DIMFLG AND VALTYP FOR RECURSION
006746' 001000  000052  20380           LHLD    DIMFLG
006747' 000000  001542'
006750' 000000  006730'
006751' 001000  000343  20400           XTHL                    ;TEXT POINTER BACK INTO (H,L)
006752' 001000  000026  20420           MVI     D,0             ;SET # DIMENSIONS =0
006753' 000000  000000
006754' 001000  000325  20440   INDLOP: PUSH    D               ;SAVE NUMBER OF DIMENSIONS
006755' 001000  000305  20460           PUSH    B               ;SAVE LOOKS
006756' 001000  003115  20480           CALL    INTIOX          ;EVALUATE INDICE INTO (D,E)
006757' 000000  003622'
006760' 000000  006747'
006761' 001000  000301  20500           POP     B               ;POP OFF THE LOOKS
006762' 001000  000361  20520           POP     PSW             ;(A) = NUMBER OF DIMENSIONS SO FAR
006763' 001000  000353  20540           XCHG                    ;(D,E)=TEXT POINTER
                                        20560                   ;(H,L)=INDICE
006764' 001000  000343  20580           XTHL                    ;PUT THE INDICE ON THE STACK
                                        20600                   ;(H,L)=VALTYP & DIMFLG
006765' 001000  000345  20620           PUSH    H               ;RESAVE VALTYP AND DIMFLG
006766' 001000  000353  20640           XCHG                    ;(H,L)=TEXT POINTER
006767' 001000  000074  20660           INR     A               ;INCREMENT # OF DIMENSIONS
006770' 001000  000127  20680           MOV     D,A             ;(D)=NUMBER OF DIMENSIONS
006771' 001000  000176  20700           MOV     A,M             ;GET TERMINATING CHARACTER
006772' 001000  000376  20720           CPI     44              ;A COMMA SO MORE INDICES FOLLOW?
006773' 000000  000054
006774' 001000  000312  20740           JZ      INDLOP          ;IF SO, READ MORE
006775' 000000  006754'
006776' 000000  006757'
006777' 001000  000317  20760           SYNCHK  ")"             ;MAKE SURE IT ENDED PROPERLY
007000' 000000  000051
007001' 001000  000042  20780           SHLD    TEMP2           ;SAVE THE TEXT POINTER
007002' 000000  001605'
007003' 000000  006775'
007004' 001000  000341  20800           POP     H               ;(H,L)= VALTYP & DIMFLG
007005' 001000  000042  20820           SHLD    DIMFLG          ;SAVE VALTYP AND DIMFLG
007006' 000000  001542'
007007' 000000  007002'
007010' 001000  000325  20840           PUSH    D               ;SAVE NUMBER OF DIMENSIONS
```

```
                                    20860   ;
                                    20880   ; AT THIS POINT [B,C]=LOOKS, THE TEXT POINTER IS IN TEMP2,
                                    20900   ; THE INDICES ARE ALL ON THE STACK, FOLLOWED BY THE NUMBER OF DIMENSIONS,
                                    20920   ;
007011' 001000  000052              20940           LHLD    ARYTAB          ;[H,L]=PLACE TO START THE SEARCH
007012' 000000  001623'
007013' 000000  007006'
007014' 001000  000076              20960           XWD     "01000,"076     ;"MVI A," AROUND THE NEXT BYTE
007015' 001000  000031              20980   LOPFDA: DAD     D               ;SKIP OVER THIS ARRAY SINCE IT'S
                                    21000                                   ;NOT THE ONE
007016' 001000  000353              21020           XCHG                    ;[D,E]=CURRENT SEARCH POINT
007017' 001000  000052              21040           LHLD    STREND          ;GET THE PLACE TO STOP INTO [H,L]
007020' 000000  001625'
007021' 000000  007012'
007022' 001000  000353              21050           XCHG                    ;[H,L]=SEARCH POINT
007023' 001000  000347              21060           COMPAR                  ;STOPPING TIME?
                                    21080   IFE     LENGTH-2,<
007024' 001000  000072              21100           LDA     VALTYP>
007025' 000000  001543'
007026' 000000  007020'
007027' 001000  000312              21120           JZ      NOTFDD          ;YES,COULDN'T FIND THIS ARRAY
007030' 000000  007104'
007031' 000000  007025'
                                    21140   IFE     LENGTH-2,<
007032' 001000  000276              21160           CMP     M               ;SEE IF THE VALTYPS ARE THE SAME
007033' 001000  000043              21180           INX     H
007034' 001000  000302              21200           JNZ     NMARY2>
007035' 000000  007050'
007036' 000000  007030'
007037' 001000  000176              21240           MOV     A,M             ;GET FIRST CHARACTER
007040' 001000  000271              21260           CMP     C               ;SEE IF IT MATCHES
007041' 001000  000043              21280           INX     H
007042' 001000  000302              21300           JNZ     NMARY1          ;NOT THIS ONE
007043' 000000  007051'
007044' 000000  007035'
007045' 001000  000176              21320           MOV     A,M             ;GET SECOND CHARACTER
007046' 001000  000270              21340           CMP     B               ;ANOTHER MATCH?
                                    21360   IFE     LENGTH-2,<
007047' 001000  000076              21380           XWD     "01000,"076     ;SKIP THIS INCREMENT WITH "MVI A,"
007050' 001000  000043              21400   NMARY2: INX     H>
007051' 001000  000043              21420   NMARY1: INX     H               ;POINT TO SIZE ENTRY
007052' 001000  000136              21440           MOV     E,M             ;[D,E]=LENGTH
007053' 001000  000043              21460           INX     H               ;OF THE ARRAY BEING LOOKED AT
007054' 001000  000126              21480           MOV     D,M
007055' 001000  000043              21500           INX     H
007056' 001000  000302              21520           JNZ     LOPFDA          ;IF NO MATCH, SKIP THIS ONE
007057' 000000  007015'
007060' 000000  007043'
                                    21540                                   ;AND TRY AGAIN
007061' 001000  000072              21560           LDA     DIMFLG          ;SEE IF CALLED BY "DIM"
007062' 000000  001542'
007063' 000000  007057'
007064' 001000  000267              21580           ORA     A               ;ZERO MEANS NO
```



```
007065' 001000  000036              21600           MVI     E,ERRDD         ;"DOUBLY DIMENSIONED" ERROR
007066' 000000  000012
007067' 001000  000302              21620           JNZ     ERROR
007070' 000000  002102'
007071' 000000  007062'
                                    21640   ;
                                    21660   ; TEMP2=THE TEXT POINTER
                                    21680   ; WE HAVE LOCATED THE VARIABLE WE WERE LOOKING FOR
                                    21700   ; AT THIS POINT [H,L] POINTS BEYOND THE SIZE TO THE NUMBER OF DIMENSIONS
                                    21720   ; THE INDICES ARE ON THE STACK FOLLOWED BY THE NUMBER OF DIMENSIONS
                                    21740   ;
007072' 001000  000361              21760           POP     PSW             ;[A]=NUMBER OF DIMENSIONS
007073' 001000  000276              21780           CMP     M               ;MAKE SURE THE NUMBER GIVEN NOW AND
                                    21800                                   ;AND WHEN THE ARRAY WAS SET UP ARE THE
                                    21820                                   ;SAME
007074' 001000  000312              21840           JZ      GETDEF          ;JUMP OFF AND READ
007075' 000000  007245'
007076' 000000  007070'
                                    21860                                   ;THE INDICES,...
007077' 001000  000036              21880   BSERR:  MVI     E,ERRBS         ;"SUBSCRIPT OUT OF RANGE"
007100' 000000  000011
007101' 001000  000303              21900           JMP     ERROR
007102' 000000  002102'
007103' 000000  007075'
                                    21920   ;
                                    21940   ; HERE WHEN VARIABLE IS NOT FOUND IN THE ARRAY TABLE
                                    21960   ;
                                    21980   ; BUILDING AN ENTRY:
                                    22000   ;
                                    22020   ;       PUT DOWN THE DESCRIPTOR
                                    22040   ;       SETUP NUMER OF DIMENSIONS
                                    22060   ;       MAKE SURE THERE IS ROOM FOR THE NEW ENTRY
                                    22080   ;       REMEMBER VARPTR
                                    22100   ;       TALLY=0 (VALTYP FOR THE EXTENDED)
                                    22120   ;       SKIP 2 LOCS FOR LATER FILL IN -- THE SIZE
                                    22140   ; LOOP: GET AN INDICE
                                    22160   ;       PUT NUMBER +1 DOWN AT VARPTR AND INCREMENT VARPTR
                                    22180   ;       TALLY= TALLY * NUMBER+1
                                    22200   ;       DECREMENT NUMBER=DIMS
                                    22220   ;       JNZ     LOOP
                                    22240   ;       CALL REASON WITH [H,L] REFLECTING LAST LOC OF VARIABLE
                                    22260   ;       UPDATE STREND
                                    22280   ;       ZERO BACKWARDS
                                    22300   ;       MAKE TALLY INCLUDE MAXDIMS
                                    22320   ;       PUT DOWN TALLY
                                    22340   ;       IF CALLED BY DIMENSION, RETURN
                                    22360   ;       OTHERWISE INDEX INTO THE VARIABLE AS IF IT
                                    22380   ;       WERE FOUND ON THE INITIAL SEARCH
                                    22400   ;
007104'                             22420   NOTFDD:
                                    22440   IFE     LENGTH-2,<
007104' 001000  000167              22460           MOV     M,A             ;PUT DOWN THE VARIABLE TYPE
007105' 001000  000043              22480           INX     H
```

```
 6019  007106' 001000  000137          22500          MOV     E,A
 6020  007107' 001000  000026          22520          MVI     D,0>            ;[D,E]=SIZE OF ONE VALUE (VALTYP)
 6021  007110' 000000  000000
 6022                                  22540  IFN     LENGTH-2,<
 6023                                  22560          LXI     D,$CODE+4>      ;INITIALIZE TALLY TO FOUR
 6024  007111' 001000  000161          22580          MOV     M,C             ;PUT DOWN THE DESCRIPTOR
 6025  007112' 001000  000043          22600          INX     H
 6026  007113' 001000  000160          22620          MOV     M,B
 6027  007114' 001000  000043          22640          INX     H
 6028  007115' 001000  000361          22660          POP     PSW             ;[A]=NUMBER OF DIMENSIONS
 6029  007116' 001000  000062          22680          STA     TEMP6           ;SETUP GETSTK CALL
 6030  007117' 000000  007124'
 6031  007120' 000000  007102'
 6032  007121' 001000  000315          22700          CALL    GETSTK          ;GET SPACE FOR DIMENSION ENTRIES
 6033  007122' 000000  002024'
 6034  007123' 000000  007117'
 6035  007124' 001000  000351          22720  TEMP6:  PCHL                    ;PLACE TO STORE NUMBER OF DIMENSIONS
 6036                                  22740                                  ;FOR GETSTK AND LATER RECALL
 6037                                  22760                                  ;!!IMPURE!! PCHL TO CONFUSE DISASSEMBLY
 6038  007125' 001000  000042          22780          SHLD    TEMP3           ;SAVE THE LOCATION TO PUT THE SIZE
 6039  007126' 000000  001575'
 6040  007127' 000000  007122'
 6041                                  22800                                  ;IN
 6042  007130' 001000  000043          22820          INX     H               ;SKIP OVER THE SIZE LOCATIONS
 6043  007131' 001000  000043          22840          INX     H
 6044  007132' 001000  000101          22860          MOV     B,C             ;[B]=NUMBER OF DIMENSIONS
 6045                                  22880                                  ;THIS DEPENDS ON THE FACT THAT GETSTK
 6046                                  22900                                  ;RETURNS ITS ARGUMENT IN (C)
 6047  007133' 001000  000160          22920          MOV     M,B             ;STORE THE NUMBER OF DIMENSIONS
 6048  007134' 001000  000043          22940          INX     H
 6049  007135' 001000  000072          22960  LOPPTA: LDA     DIMFLG          ;CALLED BY DIMENSION?
 6050  007136' 000000  001542'
 6051  007137' 000000  007126'
 6052  007140' 001000  000267          22980          ORA     A
 6053  007141' 001000  000170          23000          MOV     A,B             ;[A]=NUMBER OF DIMENSIONS
 6054  007142' 001000  000001          23020          LXI     B,$CODE+11      ;ASSUME ITS NOT "DIM"
 6055  007143' 000000  000013'
 6056  007144' 000000  007136'
 6057  007145' 001000  000312          23040          JZ      NOTDIM          ;DEFAULT DIMENSIONS TO TEN
 6058  007146' 000000  007152'
 6059  007147' 000000  007143'
 6060  007150' 001000  000301          23060          POP     B               ;POP OFF AN INDICE INTO [B,C]
 6061  007151' 001000  000003          23080          INX     B               ;ADD ONE TO IT FOR THE ZERO ENTRY
 6062  007152' 001000  000161          23100  NOTDIM: MOV     M,C             ;PUT THE MAXIMUM DOWN
 6063  007153' 001000  000043          23120          INX     H
 6064  007154' 001000  000160          23140          MOV     M,B
 6065  007155' 001000  000043          23160          INX     H
 6066  007156' 001000  000365          23180          PUSH    PSW             ;SAVE THE NUMBER OF DIMENSIONS
 6067  007157' 001000  000345          23200          PUSH    H               ;SAVE THE POINTER INTO THE NEW ENTRY
 6068  007160' 001000  000315          23220          CALL    UMULT           ;MULTIPLY [B,C]=NEWMAX BY CURTOL=[D,E]
 6069  007161' 000000  000000*
 6070  007162' 000000  007146'
 6071  007163' 001000  000353          23240          XCHG                    ;[D,E]=NEW CURTOL
```



```
 6072  007164' 001000  000341          23260          POP     H               ;GET THE POINTER INTO THE ENTRY BACK
 6073  007165' 001000  000301          23280          POP     B               ;GET THE NUMBER OF DIMENSIONS BACK
 6074  007166' 001000  000005          23300          DCR     B               ;DECREMENT THE NUMBER OF DIMENSIONS LEFT
 6075  007167' 001000  000302          23320          JNZ     LOPPTA          ;HANDLE THE OTHER INDICES
 6076  007170' 000000  007135'
 6077  007171' 000000  007161'
 6078  007172' 001000  000102          23340          MOV     B,D             ;[B,C]=SIZE
 6079  007173' 001000  000113          23360          MOV     C,E
 6080  007174' 001000  000353          23380          XCHG                    ;[D,E]=START OF VALUES
 6081  007175' 001000  000031          23400          DAD     D               ;[H,L]=END OF VALUES
 6082  007176' 001000  000332          23420          JC      BSERR           ;OUT OF MEMORY POINTER BEING GENERATED?
 6083  007177' 000000  007077'
 6084  007200' 000000  007170'
 6085  007201' 001000  000315          23440          CALL    REASON          ;SEE IF THERE IS ROOM FOR THE VALUES
 6086  007202' 000000  002045'
 6087  007203' 000000  007177'
 6088  007204' 001000  000042          23460          SHLD    STREND          ;UPDATE THE END OF STORAGE
 6089  007205' 000000  001625'
 6090  007206' 000000  007202'
 6091  007207' 001000  000053          23480  ZERITA: DCX     H               ;ZERO THE NEW ARRAY
 6092  007210' 001000  000066          23500          MVI     M,0
 6093  007211' 000000  000000
 6094  007212' 001000  000347          23520          COMPAR                  ;BACK AT THE BEGINNING?
 6095  007213' 001000  000302          23540          JNZ     ZERITA          ;NO,ZERO MORE
 6096  007214' 000000  007207'
 6097  007215' 000000  007205'
 6098  007216' 001000  000003          23560          INX     B               ;ADD ONE TO THE SIZE TO INCLUDE
 6099                                  23580                                  ;THE BYTE FOR THE NUMBER OF DIMENSIONS
 6100  007217' 001000  000147          23600          MOV     M,A             ;[M]=ZERO
 6101  007220' 001000  000072          23620          LDA     DIMFLG
 6102  007221' 000000  001542'
 6103  007222' 000000  007214'
 6104  007223' 001000  000267          23640          ORA     A               ;ARE WE DIMENSIONING ?
 6105  007224' 001000  000072          23660          LDA     TEMP6           ;GET THE NUMBER OF DIMENSIONS
 6106  007225' 000000  007124'
 6107  007226' 000000  007221'
 6108  007227' 001000  000157          23680          MOV     L,A             ;[L]=NUMBER OF DIMENSIONS
 6109  007230' 001000  000051          23700          DAD     H               ;[H,L]=NUMBER OF DIMENSIONS TIMES TWO
 6110  007231' 001000  000011          23720          DAD     B               ;ADD ON THE SIZE
 6111                                  23740                                  ;TO GET THE TOTAL NUMBER OF BYTES USED
 6112  007232' 001000  000353          23760          XCHG                    ;[D,E]=TOTAL SIZE
 6113  007233' 001000  000052          23780          LHLD    TEMP3           ;PLACE TO STORE SIZE
 6114  007234' 000000  001575'
 6115  007235' 000000  007225'
 6116  007236' 001000  000163          23800          MOV     M,E             ;PUT DOWN THE SIZE
 6117  007237' 001000  000043          23820          INX     H
 6118  007240' 001000  000162          23840          MOV     M,D
 6119  007241' 001000  000043          23860          INX     H
 6120  007242' 001000  000302          23880          JNZ     FINNOW
 6121  007245' 000000  007331'
 6122  007244' 000000  007234'
 6123                                  23900  ;
 6124                                  23920  ; AT THIS POINT [M,L] POINTS BEYOND THE SIZE TO THE NUMBER OF DIMENSIONS
```