```
                                  11720                                   ;NUMERIC CASE
                                  11740           LXI     H,DOCMP         ;COME BACK TO COMPARE
                                  11760           PUSH    H
                                  11780           JZ      FCOMP           ;DO NUMERIC COMPARE
                                  11800           XRA     A               ;SET VALUE TYPE AS NUMERIC
                                  11820           STA     VALTYP>>>
                                  11840   IFN     STRING,<
                                  11860   ;
                                  11880   ; THE FOLLOWING ROUTINE COMPARES TWO STRINGS
                                  11900   ; ONE WITH DESC IN [D,E] OTHER wITH DESC, IN [FACLO, FACLO+1]
                                  11920   ; A=0 IF STRINGS EQUAL
                                  11940   ; A=377 IF B,C,D,E POINTER FACLO
                                  11960   ; A=1 IF B,C,D,E .LT. FACLO
                                  11980   ;
                                  12000   IFN     LENGTH-2,<
                                  12020   STRCMP: PUSH    D               ;SAVE DESC, POINTER TO FIRST STR,
                                  12040           CALL    FREFAC          ;FREE THE FACLO STR
                                  12060           POP     D               ;RESTORE 1ST DESC, POINTER
                                  12080           PUSHM                   ;SAVE LENGTH
                                  12100           PUSHM                   ;SAVE POINTER
                                  12120           CALL    FRETMP          ;FREES IST DESC, POINTER
                                  12140           CALL    MOVRM           ;[B,C] POINT AT FIRST CHAR
                                  12160                                   ;[E] HAS THE LENGTH
                                  12180           POP     H               ;GET 2ND CHARACTER POINTER IN H
                                  12200           XTHL                    ;GET 2ND CHARACTER COUNTER IN L
                                  12220           MOV     D,L>            ;SAVE IN D
                                  12240   IFE     LENGTH-2,<
006320' 001000  000315            12260   STRCMP: CALL    FRESTR          ;FREE UP THE FAC STRING, AND GET THE
006321' 000000  010431'
006322' 000000  006310'
                                  12280                                   ;POINTER TO THE FAC DESCRIPTOR IN [H,L]
006323' 001000  000176            12300           MOV     A,M             ;SAVE THE LENGTH OF THE FAC STRING IN [A]
006324' 001000  000043            12320           INX     H
006325' 001000  000116            12340           MOV     C,M             ;SAVE THE POINTER AT THE FAC STRING
                                  12342                                   ;DATA IN [B,C]
006326' 001000  000043            12360           INX     H
006327' 001000  000106            12380           MOV     B,M
006330' 001000  000321            12400           POP     D               ;GET THE STACK STRING POINTER
006331' 001000  000305            12420           PUSH    B               ;SAVE THE POINTER AT THE FAC STRING DATA
006332' 001000  000365            12440           PUSH    PSW             ;SAVE THE FAC STRING LENGTH
006333' 001000  000315            12460           CALL    FRETMP          ;FREE UP THE STACK STRING AND RETURN
006334' 000000  010440'
006335' 000000  006321'
                                  12480                                   ;THE POINTER TO THE STACK STRING DESCRIPTOR
                                  12500                                   ;IN [H,L]
006336' 001000  000321            12520           POP     D               ;[D]=LENGTH OF FAC STRING
006337' 001000  000136            12540           MOV     E,M             ;[E]=LENGTH OF STACK STRING
006340' 001000  000043            12560           INX     H
006341' 001000  000116            12580           MOV     C,M             ;[B,C]=POINTER AT STACK STRING
006342' 001000  000043            12600           INX     H
006343' 001000  000106            12620           MOV     B,M>
006344' 001000  000341            12640           POP     H               ;GET BACK 2ND CHARACTER POINTER
006345' 001000  000173            12660   CSLOOP: MOV     A,E             ;BOTH STRINGS ENDED
```

```
006346' 001000  000262            12680           ORA     D               ;TEST BY OR'ING THE LENGTHS TOGETHER
006347' 001000  000310            12700           RZ                      ;IF SO, RETURN WITH A ZERO
006350' 001000  000172            12720           MOV     A,D             ;GET FACLO STRING LENGTH
006351' 001000  000267            12740           ORA     A               ;IF IT ENDED, OTHER MUST NOT HAVE
006352' 001000  000057            12760           CMA                     ;MAKE -1
006353' 001000  000310            12780           RZ                      ;TEST
006354' 001000  000257            12800           XRA     A               ;MUST NOT HAVE BEEN ZERO, TEST CASE
006355' 001000  000273            12820           CMP     E               ;OF B,C,D,E STRING HAVING ENDED FIRST
006356' 001000  000074            12840           INR     A               ;RETURN WITH A=1
006357' 001000  000320            12860           RNC                     ;TEST THE CONDITION
                                  12880   ;HERE WHEN NEITHER STRING ENDED
006360' 001000  000025            12900           DCR     D               ;DECREMENT BOTH CHARACTER COUNTS
006361' 001000  000035            12920           DCR     E
006362' 001000  000012            12940           LDAX    B               ;GET CHARACTER FROM B,C,D,E STRING
006363' 001000  000276            12960           CMP     M               ;COMPARE WITH FACLO STRING
006364' 001000  000043            12980           INX     H               ;BUMP POINTERS (INX DOESNT CLOBBER CC'S)
006365' 001000  000003            13000           INX     B
006366' 001000  000312            13020           JZ      CSLOOP          ;IF BOTH THE SAME, MUST BE MORE TO STRINGS
006367' 000000  006345'
006370' 000000  006334'
006371' 001000  000077            13040           CMC                     ;HERE WHEN STRINGS DIFFER
006372' 001000  000303            13060           JMP     SIGNS>          ;SET [A] ACCORDING TO CARRY
006373' 000000  000000*
006374' 000000  006367'
                                  13080   IFN     LENGTH,<
006375' 001000  000074            13100   DOCMP:  INR     A               ;SETUP BITS
006376' 001000  000217            13120           ADC     A               ;4=LESS 2=EQUAL 1=GREATER
006377' 001000  000301            13140           POP     B               ;WHAT DID HE WANT?
006400' 001000  000240            13160           ANA     B               ;ANY BITS MATCH?
006401' 001000  000306            13180           ADI     255             ;MAP 0 TO 0
006402' 000000  000377
006403' 001000  000237            13200           SBB     A               ;AND ALL OTHERS TO 377
                                  13220   IFE     LENGTH-2,<
006404' 001000  000315            13240           CALL    CONIA##         ;CONVERT [A] TO AN INTEGER SIGNED
006405' 000000  000000*
006406' 000000  006373'
006407' 001000  000303            13260           JMP     RETAOP>         ;RETURN FROM OPERATOR APPLICATION
006410' 000000  005354'
006411' 000000  006405'
                                  13280   IFN     LENGTH-2,<
                                  13300           JMP     FLOAT           ;MAKE FAC=[A] SIGNED
                                  13320                                   ;COULD FALL INTO FLOAT BUT MESSY [SAVES
                                  13340                                   ;TWO BYTES]
                                  13360
                                  13380   NOTER:  MVI     D,90            ;"NOT" HAS PRECEDENCE 90
                                  13400           CALL    LPOPER          ;GO PERFORM
                                  13420   IFN     STRING,<CALL    CHKNUM> ;MAKE SURE ITS INT
                                  13440           CALL    DEINT           ;GET VALUE IN [D,E]
                                  13460           MOV     A,E
                                  13480           CMA                     ;COMPLEMENT
                                  13500           MOV     C,A             ;[C] LOW ORDER OF ANSWER
                                  13520           MOV     A,D
                                  13540           CMA                     ;COMPLEMENT HIGH ORDER TOO
```

Formula Evaluation

```
                                      CALL    GIVACF          ;FLOAT [A,C] AS RESULT
                                      POP     B               ;TAKE RETURN ADDRESS OF FRMEVL
                                      JMP     RETAOP>>        ;OFF AND RETURN TO THE RIGHT
                                                              ;PLACE SO THE TEXT POINTER
                                                              ;WILL GET SET UP TO WHAT IT WAS
                                                              ;WHEN LPOPER RETURNED.
                              IFE     LENGTH-2,<
006412' 001000 000026         NOTER:  MVI     D,90            ;"NOT" HAS PRECEDENCE 90, SO
006413' 000000 000132
006414' 001000 000315                 CALL    LPOPER          ;FORMULA EVALUATION IS ENTERED WITH A DUMMY
006415' 000000 005341'
006416' 000000 006410'
                                                              ;ENTRY OF 90 ON THE STACK
006417' 001000 000315                 CALL    FRCINT          ;COERCE THE ARGUMENT TO INTEGER
006420' 000000 005573*
006421' 000000 006415'
006422' 001000 000175                 MOV     A,L             ;COMPLEMENT [H,L]
006423' 001000 000057                 CMA
006424' 001000 000157                 MOV     L,A
006425' 001000 000174                 MOV     A,H
006426' 001000 000057                 CMA
006427' 001000 000147                 MOV     H,A
006430' 001000 000042                 SHLD    FACLO           ;UPDATE THE FAC
006431' 000000 001637'
006432' 000000 006420'
006433' 001000 000301                 POP     B               ;FRMEVL, AFTER SEEING THE PRECEDENCE
                                                              ;OF 90 THINKS IT IS APPLYING AN OPERATOR
                                                              ;SO IT HAS THE TEXT POINTER IN TEMP2 SO

006434' 001000 000303                 JMP     RETAOP          ;RETURN TO REFETCH IT
006435' 000000 005354'
006436' 000000 006431'
                              ;
                              ; DANDOR APPLIES THE "AND" AND "OR" OPERATORS
                              ; AND SHOULD BE USED TO IMPLEMENT ALL LOGICAL OPERATORS.
                              ; WHENEVER AN OPERATOR IS APPLIED, ITS PRECEDENCE IS IN [B].
                              ; THIS FACT IS USED TO DISTINGUISH BETWEEN "AND" AND "OR".
                              ; THE RIGHT HAND ARGUMENT IS COERCED TO INTEGER, JUST AS
                              ; THE LEFT HAND ONE WAS WHEN IT WAS PUSHED ON THE STACK.
                              ;
006437' 001000 000305         DANDOR: PUSH    B               ;SAVE THE PRECEDENCE "OR"=70
006440' 001000 000315                 CALL    FRCINT          ;COERCE RIGHT HAND ARGUMENT TO INTEGER
006441' 000000 006420*
006442' 000000 006435'
006443' 001000 000361                 POP     PSW             ;GET BACK THE PRECEDENCE TO DISTINGUISH
                                                              ;"AND" AND "OR"
006444' 001000 000321                 POP     D               ;POP OFF THE LEFT HAND ARGUMENT
006445' 001000 000376                 CPI     70              ;SET ZERO FOR "OR"
006446' 000000 000106
006447' 001000 000173                 MOV     A,E             ;SETUP LOW IN [A]
006450' 001000 000312                 JZ      ORFIN           ;DO "OR" IF PRECEDENCE WAS 70
006451' 000000 006463'
006452' 000000 006441'
```

```
006453' 001000 000245                 ANA     L
006454' 001000 000157                 MOV     L,A
006455' 001000 000174                 MOV     A,H
006456' 001000 000242                 ANA     D
006457' 001000 000147                 MOV     H,A
006460' 001000 000303                 JMP     MAKINT          ;RETURN THE INTEGER [H,L]
006461' 000000 000006*
006462' 000000 006451'
                                                              ; AS THE "AND"ED RESULT
006463' 001000 000265         ORFIN:  ORA     L
006464' 001000 000157                 MOV     L,A
006465' 001000 000174                 MOV     A,H
006466' 001000 000262                 ORA     D
006467' 001000 000147                 MOV     H,A
006470' 001000 000303                 JMP     MAKINT>         ;RETURN THE INTEGER [H,L]
006471' 000000 006461*
006472' 000000 006461'
                                                              ; AS THE "OR"ED RESULT
                              PAGE
```

Formula Evaluation

```
   5496                                    14360   SUBTTL  DIMENSION & VARIABLE SEARCHING
   5497   006473' 001000  000053           14380   DIMCON: DCX     H                   ;SEE IF COMMA ENDED THIS VARIABLE
   5498   006474' 001000  000327           14400           CHRGET
   5499   006475' 001000  000310           14420           RZ                          ;IF TERMINATOR, GOOD BYE
   5500   006476' 001000  000317           14440           SYNCHK  44                  ;MUST BE COMMA
   5501   006477' 000000  000054
   5502                                    14442   ;
   5503                                    14444   ; THE "DIM" CODE SETS DIMFLG AND THEN FALLS INTO THE VARIABLE
   5504                                    14446   ; SEARCH ROUTINE, THE VARIABLE SEARCH ROUTINE LOOKS AT
   5505                                    14448   ; DIMFLG AT THREE DIFFERENT POINTS:
   5506                                    14450   ;
   5507                                    14452   ;       1) IF AN ENTRY IS FOUND, DIMFLG BEING ON INDICATES
   5508                                    14454   ;               A "DOUBLY DIMENSIONED" VARIABLE
   5509                                    14456   ;       2) WHEN A NEW ENTRY IS BEING BUILT DIMFLG'S BEING ON
   5510                                    14458   ;               INDICATES THE INDICES SHOULD BE USED FOR
   5511                                    14460   ;               THE SIZE OF EACH INDICE, OTHERWISE THE DEFAULT
   5512                                    14462   ;               OF TEN IS USED,
   5513                                    14464   ;       3) WHEN THE BUILD ENTRY CODE FINISHES, ONLY IF DIMFLG IS
   5514                                    14466   ;               OFF WILL INDEXING BE DONE
   5515                                    14468   ;
   5516   006500' 001000  000001           14478   DIM:    LXI     B,DIMCON            ;PLACE TO COME BACK TO
   5517   006501' 000000  006473'
   5518   006502' 000000  006471'
   5519   006503' 001000  000305           14480           PUSH    B
   5520   006504' 001000  000366           14500           XWD     ^O1000,^O366        ;"ORI" NON ZERO THING
   5521                                    14520   ;
   5522                                    14540   ; ROUTINE TO READ THE VARIABLE NAME AT THE CURRENT TEXT POSITION
   5523                                    14560   ; AND PUT A POINTER TO ITS VALUE IN [D,E], [H,L] IS UPDATED
   5524                                    14580   ; TO POINT TO THE CHARACTER AFTER THE VARIABLE NAME,
   5525                                    14600   ; VALTYP IS SETUP, NOTE THAT EVALUATING SUBSCRIPTS IN
   5526                                    14620   ; A VARIABLE NAME CAN CAUSE RECURSIVE CALLS TO PTRGET SO AT
   5527                                    14640   ; THAT POINT ALL VALUES MUST BE STORED ON THE STACK,
   5528                                    14660   ;
   5529   006505' 001000  000257           14680   PTRGET: XRA     A                   ;MAKE [A]=0
   5530   006506' 001000  000062           14700           STA     DIMFLG              ;FLAG IT AS SUCH
   5531   006507' 000000  001542'
   5532   006510' 000000  006501'
   5533   006511' 001000  000106           14720           MOV     B,M                 ;GET FIRST CHARACTER IN [B]
   5534   006512' 001000  000315           14740   PTRGT2: CALL    ISLET               ;CHECK FOR LETTER
   5535   006513' 000000  003612'
   5536   006514' 000000  006507'
   5537   006515' 001000  000332           14760           JC      SNERR               ;MUST HAVE A LETTER
   5538   006516' 000000  002072'
   5539   006517' 000000  006513'
   5540   006520' 001000  000257           14780           XRA     A
   5541   006521' 001000  000117           14800           MOV     C,A                 ;ASSUME NO SECOND CHARACTER
   5542                                    14820   IFN     LENGTH-2,<
   5543                                    14840   IFN     STRING,<
   5544                                    14860           STA     VALTYP>>            ;DEFAULT IS ZERO (NUMERIC)
   5545   006522' 001000  000327           14880           CHRGET                      ;GET THE FOLLOWING CHARACTER
   5546                                    14900   IFE     LENGTH,<
   5547                                    14920           JNC     NOSEC>              ;ONLY NUMBERS ALLOWED
   5548                                    14940   IFN     LENGTH,<
```

FORMULA EVALUATION

```
   5549   006523' 001000  000332           14960           JC      ISSEC               ;CARRY SET BY CHRGET IF CHARACTER IS
   5550   006524' 000000  006534'
   5551   006525' 000000  006516'
   5552                                    14980                                       ;NUMERIC
   5553   006526' 001000  000315           15000           CALL    ISLET               ;SET CARRY IF NOT ALPHABETIC
   5554   006527' 000000  003612'
   5555   006530' 000000  006524'
   5556   006531' 001000  000332           15020           JC      NOSEC>              ;ALLOW ALPHABETICS
   5557   006532' 000000  006547'
   5558   006533' 000000  006527'
   5559   006534' 001000  000117           15040   ISSEC:  MOV     C,A                 ;IT IS A NUMBER--SAVE IN C
   5560   006535' 001000  000327           15060   EATEM:  CHRGET                      ;LOOK AT NEXT CHARACTER
   5561                                    15080   IFN LENGTH,<
   5562   006536' 001000  000332           15100           JC      EATEM               ;SKIP NUMERICS
   5563   006537' 000000  006535'
   5564   006540' 000000  006532'
   5565   006541' 001000  000315           15120           CALL    ISLET
   5566   006542' 000000  003612'
   5567   006543' 000000  006537'
   5568   006544' 001000  000322           15140           JNC     EATEM>              ;SKIP ALPHABETICS
   5569   006545' 000000  006535'
   5570   006546' 000000  006542'
   5571   006547'                          15160   NOSEC:
   5572                                    15180   IFE     LENGTH-2,<
   5573   006547' 001000  000021           15200           LXI     D,HAVTYP            ;SAVE JUMPS BY USING RETURN ADDRESS
   5574   006550' 000000  006574'
   5575   006551' 000000  006545'
   5576   006552' 001000  000325           15220           PUSH    D
   5577   006553' 001000  000026           15240           MVI     D,8                 ;ASSUME ITS DOUBLE PRECISION
   5578   006554' 000000  000010
   5579   006555' 001000  000376           15260           CPI     "#"                 ;CHECK THE CHARACTER
   5580   006556' 000000  000043
   5581   006557' 001000  000310           15280           RZ                          ;WHEN WE MATCH, SETUP VALTYP
   5582   006560' 001000  000026           15300           MVI     D,2                 ;CHECK FOR INTEGER
   5583   006561' 000000  000002
   5584   006562' 001000  000376           15320           CPI     "%"
   5585   006563' 000000  000045
   5586   006564' 001000  000310           15340           RZ
   5587   006565' 001000  000024           15360           INR     D                   ;CHECK FOR STRING
   5588   006566' 001000  000376           15380           CPI     "$"
   5589   006567' 000000  000044
   5590   006570' 001000  000310           15400           RZ
   5591   006571' 001000  000024           15420           INR     D                   ;SINGLE PRECISION IS THE DEFAULT
   5592   006572' 001000  000053           15440           DCX     H                   ;NO MARKING CHARACTER
   5593   006573' 001000  000311           15460           RET                         ;GET RID OF RETURN ADDRESS
   5594   006574' 001000  000172           15480   HAVTYP: MOV     A,D                 ;SETUP VALTYP
   5595   006575' 001000  000062           15500           STA     VALTYP
   5596   006576' 000000  001543'
   5597   006577' 000000  006550'
   5598   006600' 001000  000327           15520           CHRGET>                     ;READ PAST TYPE MARKER
   5599                                    15540   IFN     LENGTH-2,<
   5600                                    15560   IFN     STRING,<
   5601                                    15580           SUI     "$"                 ;IS IT A STRING?
```

```
  5602                              15600            JNZ     NOTSTR          ;IF NOT VALTYP ALREADY=0
  5603                              15620            INR     A               ;[A]=1
  5604                              15640            STA     VALTYP          ;FLAG THIS AS A STRING
  5605                              15660            RRC                     ;MAKE [A]=128
  5606                              15680            ADD     C               ;MAKE [A]=SECOND CHARACTER
  5607                              15700            MOV     C,A             ;BACK INTO [C] WITH STRING BIT ON
  5608                              15720            CHRGET                  ;GET CHARACTER AFTER "$"
  5609                              15740    NOTSTR: >>
  5610                              15760    IFN     LENGTH,<
  5611  006601' 001000  000072      15780            LDA     SUBFLG          ;GET FLAG WHETHER TO ALLOW ARRAYS
  5612  006602' 000000  001601'
  5613  006603' 000000  006576'
  5614  006604' 001000  000206      15800            ADD     M>              ;ADD ONTO CHARACTER
  5615                              15820            CPI     "("             ;ARRAY PERHAPS (IF SUBFLG SET NEVER WILL MATCH)
  5616  006605' 001000  000376
  5617  006606' 000000  000050
  5618  006607' 001000  000312      15840            JZ      ISARY           ;IT IS!
  5619  006610' 000000  006745'
  5620  006611' 000000  006602'
  5621                              15860    IFN     LENGTH,<
  5622  006612' 001000  000257      15880            XRA     A               ;ALLOW PARENS AGAIN
  5623  006613' 001000  000062      15900            STA     SUBFLG>         ;SAVE IN FLAG LOCATION
  5624  006614' 000000  001601'
  5625  006615' 000000  006610'
  5626  006616' 001000  000345      15920            PUSH    H               ;SAVE THE TEXT POINTER
  5627                              15960    IFE     LENGTH-2,<
  5628  006617' 001000  000172      15980            MOV     A,D             ;VALUE TYPE INTO [A]
  5629  006620' 001000  000052      15990            LHLD    VARTAB          ;[H,L]=PLACE TO START THE SEARCH
  5630  006621' 000000  001621'
  5631  006622' 000000  006614'
  5632  006623' 001000  000365      16000    LOPFND: PUSH    PSW             ;SAVE THE VALUE TYPE
  5633  006624' 001000  000353      16020            XCHG                    ;[D,E]=POINTER INTO SIMPLE VARIABLES
  5634  006625' 001000  000052      16040            LHLD    ARYTAB          ;[H,L]=END OF SIMPLE VARIABLES
  5635  006626' 000000  001623'
  5636  006627' 000000  006621'
  5637  006630' 001000  000347      16060            COMPAR                  ;SEE IF THE END HAS BEEN REACHED
  5638  006631' 001000  000341      16080            POP     H               ;[H]=VALTYP
  5639  006632' 001000  000312      16100            JZ      NOTFNS          ;COULDN'T FIND IT, SO MAKE A NEW ENTRY
  5640  006633' 000000  006671'
  5641  006634' 000000  006626'
  5642  006635' 001000  000032      16120            LDAX    D               ;GET THE VALTYP OF THIS SIMPLE VARIABLE
  5643  006636' 001000  000157      16140            MOV     L,A             ;SAVE SO WE KNOW HOW MUCH TO SKIP
  5644  006637' 001000  000274      16160            CMP     H               ;COMPARE WITH OUR VALTYP
  5645  006640' 001000  000023      16180            INX     D
  5646  006641' 001000  000302      16200            JNZ     NOTIT1          ;NOT RIGHT KIND -- SKIP IT
  5647  006642' 000000  006660'
  5648  006643' 000000  006633'
  5649  006644' 001000  000032      16220            LDAX    D               ;[A]=FIRST CHARACTER OF THIS VARIABLE
  5650  006645' 001000  000271      16240            CMP     C               ;SEE IF OUR VARIABLE MATCHES
  5651  006646' 001000  000302      16260            JNZ     NOTIT1
  5652  006647' 000000  006660'
  5653  006650' 000000  006642'
  5654  006651' 001000  000023      16280            INX     D
```

```
  5655  006652' 001000  000032      16300            LDAX    D               ;SEE IF SECOND CHACRACTER MATCHES
  5656  006653' 001000  000270      16320            CMP     B
  5657  006654' 001000  000312      16340            JZ      FINPTR          ;THAT WAS IT, ALL DONE
  5658  006655' 000000  006742'
  5659  006656' 000000  006647'
  5660  006657' 001000  000076      16360            XWD     ¯01000,¯076     ;"MVI A," AROUND THIS INX SINCE THE POINTER
  5661                              16380                                    ;IS ALREADY INCREMENTED
  5662  006660' 001000  000023      16400    NOTIT1: INX     D
  5663  006661' 001000  000023      16420            INX     D
  5664  006662' 001000  000174      16440            MOV     A,H             ;SKIP OVER THE
  5665                              16460                                    ;CURRENT VARIABLE SINCE WE DIDN'T MATCH
  5666                              16480                                    ;SAVE THE VALTYP IN [A]
  5667  006663' 001000  000046      16500            MVI     H,0             ;[H,L]=NUMBER OF BYTES TO SKIP
  5668  006664' 000000  000000
  5669  006665' 001000  000031      16520            DAD     D               ;ADD ON THE POINTER
  5670  006666' 001000  000303      16540            JMP     LOPFND          ;AND SEARCH SOME MORE
  5671  006667' 000000  006623'
  5672  006670' 000000  006655'
  5673  006671' 001000  000305      16560    NOTFNS: PUSH    B               ;SAVE THE LOOKS
  5674  006672' 001000  000114      16580            MOV     C,H             ;[B,C]=LENGTH OF THIS VARIABLE
  5675  006673' 001000  000107      16600            MOV     B,A             ;[B]=0
  5676  006674' 001000  000305      16620            PUSH    B               ;SAVE THE VALTYP ON THE STACK
  5677  006675' 001000  000003      16640            INX     B               ;MAKE THE LENGTH INCLUDE
  5678                              16660                                    ;THE LOOKS TOO
  5679  006676' 001000  000003      16680            INX     B
  5680  006677' 001000  000003      16700            INX     B>
  5681                              16720    IFN     LENGTH-2,<
  5682                              16730            LHLD    ARYTAB          ;PLACE TO STOP SEARCHING
  5683                              16740            XCHG
  5684                              16760            LHLD    VARTAB          ;GET THE PLACE TO START
  5685                              16780    LOPFND: COMPAR                  ;SEE IF WE ARE THERE
  5686                              16800            JZ      NOTFNS          ;COULDN'T FIND THIS VARIABLE
  5687                              16820                                    ;SO MAKE ROOM FOR IT
  5688                              16840            MOV     A,C
  5689                              16860            SUB     M               ;IS THIS VARIABLE THE ONE?
  5690                              16880            INX     H
  5691                              16900            JNZ     NOTIT           ;NOPE
  5692                              16920            MOV     A,B
  5693                              16940            SUB     M               ;TRY SECOND CHARACTER MATCHING
  5694                              16960    NOTIT:  INX     H
  5695                              16980            JZ      FINPTR          ;THAT WAS IT!
  5696                              17000            INX     H               ;SKIP OVER THAT ONE--NOT IT
  5697                              17020            INX     H
  5698                              17040            INX     H
  5699                              17060            INX     H
  5700                              17080            JMP     LOPFND          ;TRY AGAIN
  5701                              17100    NOTFNS: PUSH    B               ;REMEMBER WHAT THIS
  5702                              17120                                    ;VARIABLE LOOKS LIKE
  5703                              17140            LXI     B,6+$CODE>      ;THE AMOUNT TO SHOVE
  5704                              17160                                    ;EVERYTHING UP BY
  5705  006700' 001000  000052      17180            LHLD    STREND          ;THE CURRENT END OF STORAGE
  5706  006701' 000000  001625'
  5707  006702' 000000  006667'
```

Formula Evaluation