```
        SBB     H               ;NEGATE HO
        MOV     H,A             ;SAVE IT
        RET                     ;ALL DONE


        ;INTEGER ABSOLUTE VALUE
        ;ALTERS A,B,C,D,E,H,L
IABS:   LDA     FACLO+1         ;GET SIGN OF INTEGER IN FAC
        ORA     A               ;CHECK ITS SIGN
        RP                      ;IT IS POSITIVE, LEAVE IT ALONE
                                ;FALL INTO INEG AND NEGATE IT


        ;INTEGER NEGATION
        ;ALTERS A,B,C,D,E,H,L
INEG:   LHLD    FACLO           ;GET THE INTEGER
        CALL    INEGHL          ;NEGATE IT
        SHLD    FACLO           ;STORE IT BACK IN THE FAC
        XRI     200             ;CHECK FOR SPECIAL CASE OF 32768
        ORA     L
        RNZ                     ;IT DID NOT OCCUR, EVERYTHING IS FINE
        XCHG                    ;WE HAVE IT, FLOAT 32768
        MVI     A,4             ;CHANGE VALTYP TO "SINGLE PRECISION"
        STA     VALTYP
INEGAD: MVI     B,230           ;ENTRY FROM IADD, SET EXPONENT
        JMP     FLOATR          ;GO FLOAT THE NUMBER


        ;MOD OPERATOR
        ;(HL):=(DE)-(DE)/(HL)*(HL),  (DE)=QUOTIENT
        ;ALTERS A,B,C,D,E,H,L
MOD:    PUSH    D               ;SAVE (DE) FOR ITS SIGN
        CALL    IDIV            ;DIVIDE AND GET THE REMAINDER
        XCHG                    ;PUT REMAINDER IN (DE)
        MVI     A,2             ;SET VALTYP TO "INTEGER" IN CASE RESULT OF
        STA     VALTYP          ; THE DIVISION WAS 32768
        POP     PSW             ;GET THE SIGN OF (DE) BACK
        JMP     INEGA>          ;NEGATE THE REMAINDER IF NECESSARY
PAGE
```

```
SUBTTL  DOUBLE PRECISION ARITHMETIC ROUTINES
IFE     LENGTH-2,<
COMMENT %
        DOUBLE PRECISION ARITHMETIC CONVENTIONS

DOUBLE PRECISION NUMBERS ARE 8 BYTE QUANTITIES
THE LAST 4 BYTES IN MEMORY ARE IN THE SAME FORMAT AS SINGLE PRECISION NUMBERS
THE FIRST 4 BYTES ARE 32 MORE LOW ORDER BITS OF PRECISION
THE LOWEST ORDER BYTE COMES FIRST IN MEMORY

CALLING CONVENTIONS:
FOR ONE ARGUMENT FUNCTIONS:
        THE ARGUMENT IS IN THE FAC, THE RESULT IS LEFT IN THE FAC
FOR TWO ARGUMENT OPERATIONS:
        THE FIRST ARGUMENT IS IN ARG-7,6,5,4,3,2,1,0  (NOTE: ARGLO=ARG-7)
        THE SECOND ARGUMENT IS IN THE FAC
        THE RESULT IS LEFT IN THE FAC
VALTYP(DOUBLE PRECISION)=10 OCTAL
%


        ;DOUBLE PRECISION SUBTRACTION    FAC:=ARG-FAC
        ;ALTERS ALL REGISTERS
DSUB:   CALL    NEG             ;NEGATE THE SECOND ARGUMENT
                                ;FALL INTO DADD


        ;DOUBLE PRECISION ADDITION       FAC:=ARG+FAC
        ;ALTERS ALL REGISTERS
DADD:   LXI     H,ARG           ;GET  POINTER TO EXPONENT OF FIRST ARGUMENT
        MOV     A,M             ;CHECK IF IT IS ZERO
        ORA     A
        RZ                      ;IT IS, RESULT IS ALREADY IN FAC
        MOV     B,A             ;SAVE EXPONENT FOR UNPACKING
        DCX     H               ;POINT TO HO AND SIGN
        MOV     C,M             ;GET HO AND SIGN FOR UNPACKING
        LXI     D,FAC           ;GET POINTER TO EXPONENT OF SECOND ARGUMENT
        LDAX    D               ;GET EXPONENT
        ORA     A               ;SEE IF IT IS ZERO
        JZ      VMOVFA          ;IT IS, MOVE ARG TO FAC AND WE ARE DONE
        SUB     B               ;SUBTRACT EXPONENTS TO GET SHIFT COUNT
        JNC     DADD2           ;PUT THE SMALLER NUMBER IN FAC
        CMA                     ;NEGATE SHIFT COUNT
        INR     A
        PUSH    PSW             ;SAVE SHIFT COUNT
        PUSH    B               ;SAVE HO TO UNPACK LATER
        MVI     C,10            ;SWITCH FAC AND ARG, SET UP A COUNT
        INX     H               ;POINT TO ARG
DADD1:  LDAX    D               ;GET A BYTE OF THE FAC
        MOV     B,M             ;GET A BYTE OF ARG
        MOV     M,A             ;PUT THE FAC BYTE IN ARG
        MOV     A,B             ;PUT THE ARG BYTE IN A
        STAX    D               ;PUT THE ARG BYTE IN FAC
        DCX     D               ;POINT TO THE NEXT LO BYTE OF FAC
        DCX     H               ;POINT TO THE NEXT LO BYTE OF ARG
        DCR     C               ;ARE WE DONE?
        JNZ     DADD1           ;NO, DO THE NEXT LO BYTE
        POP     B               ;GET THE HO BACK
        POP     PSW             ;GET THE SHIFT COUNT BACK
DADD2:  CPI     71              ;ARE WE WITHIN 56 BITS?
        RNC                     ;NO, ALL DONE
        PUSH    PSW             ;SAVE SHIFT COUNT
        CALL    UNPACK          ;UNPACK THE NUMBERS
        MOV     B,A             ;SAVE SUBTRACTION FLAG
        MOV     A,C             ;SAVE THE UNPACKED HO
        STA     ARG-1
        POP     PSW             ;GET SHIFT COUNT
        CALL    DSHFTR          ;SHIFT FAC RIGHT THE RIGHT NUMBER OF TIMES
        ORA     B               ;GET SUBTRACTION FLAG, HERE A=0
        JP      DADD3           ;SUBTRACT NUMBERS IF THEIR SIGNS ARE DIFFERENT
        CALL    DADDAA          ;SIGNS ARE THE SAME, ADD THE NUMBERS
        JNC     DROUND          ;ROUND THE RESULT IF NO CARRY
        INR     M               ;WE HAVE OVERFLOW, ADD ONE TO THE EXPONENT
        JZ      OVERR           ;CHECK FOR OVERFLOW
        MVI     D,1             ;SHIFT NUMBER RIGHT ONE, SHIFT IN CARRY
        CALL    DSHFRA
        JMP     DROUND          ;ROUND THE RESULT
DADD3:  XWD     1000,076        ;"MVI   A", SUBTRACT THE NUMBERS
        SBB     M               ;GET THE SUBTRACT INSTRUCTION IN A
        CALL    DADDA           ;SUBTRACT THE NUMBERS
        INX     H               ;POINT TO THE UNPACKED SIGN
        MOV     A,M             ;COMPLEMENT IT, SINCE THE FAC WAS SMALLER
        CMA
        MOV     M,A
        CC      DNEGR           ;NEGATE THE RESULT IF IT WAS NEATIVE
                                ;FALL INTO DNORML


        ;NORMALIZE FAC
        ;ALTERS A,B,C,D,H,L
DNORML: XRA     A               ;CLEAR SHIFT COUNT
DNORM1: MOV     B,A             ;SAVE SHIFT COUNT
        LDA     FAC-1           ;GET HO
        ORA     A               ;SEE IF WE CAN SHIFT 8 LEFT
        JNZ     DNORM5          ;WE CAN'T, SEE IF NUMBER IS NORMALIZED
        LXI     H,DFACLO-1      ;WE CAN, GET POINTER TO LO
        MVI     C,10            ;SET UP A COUNT
DNORM2: MOV     D,M             ;GET A BYTE OF FAC
        MOV     M,A             ;PUT IN BYTE FROM LAST LOCATION, THE FIRST
                                ; TIME THROUGH A IS ZERO
        MOV     A,D             ;PUT THE CURRENT BYTE IN A FOR NEXT TIME
        INX     H               ;INCREMENT POINTER TO NEXT HIGHER ORDER
        DCR     C               ;ARE WE DONE?
        JNZ     DNORM2          ;NO, DO THE NEXT BYTE
        MOV     A,B             ;SUBTRACT 8 FROM SHIFT COUNT
        SUI     10
```

```
        CPI     300             ;HAVE WE SHIFTED ALL BYTES TO ZERO?
        JNZ     DNORM1          ;NO, TRY TO SHIFT 8 MORE
        JMP     ZERO            ;YES, THE NUMBER IS ZERO
DNORM3: DCR     B               ;DECREMENT SHIFT COUNT
        LXI     H,DFACLO-1      ;GET POINTER TO LO
        CALL    DSHFLC          ;SHIFT THE FAC LEFT
        ORA     A               ;SEE IF NUMBER IS NORMALIZED
DNORM5: JP      DNORM3          ;SHIFT FAC LEFT ONE IF IT IS NOT NORMALIZED
        MOV     A,B             ;GET THE SHIFT COUNT
        ORA     A               ;SEE IF NO SHIFTING WAS DONE
        JZ      DROUND          ;NONE WAS, PROCEED TO ROUND THE NUMBER
        LXI     H,FAC           ;GET POINTER TO EXPONENT
        ADD     M               ;UPDATE IT
        MOV     M,A             ;SAVE UPDATED EXPONENT
        JNC     ZERO            ;UNDERFLOW, THE RESULT IS ZERO
        RZ                      ;RESULT IS ALREADY ZERO, WE ARE DONE
                                ;FALL INTO DROUND AND ROUND THE RESULT


        ;ROUND FAC
        ;ALTERS A,B,H,L
DROUND: LDA     DFACLO-1        ;GET EXTRA BYTE TO SEE IF WE HAVE TO ROUND
DROUNB: ORA     A               ;ENTRY FROM DDIV
        CM      DROUNA          ;ROUND UP IF NECESSARY
        LXI     H,FAC+1         ;GET POINTER TO UNPACKED SIGN
        MOV     A,M             ;GET SIGN
        ANI     200             ;ISOLATE SIGN BIT
        DCX     H               ;POINT TO HO
        DCX     H
        XRA     M               ;PACK SIGN AND HO
        MOV     M,A             ;PUT PACKED SIGN AND HO IN FAC
        RET                     ;WE ARE DONE


        ;SUBROUTINE FOR ROUND: ADD ONE TO FAC
DROUNA: LXI     H,DFACLO        ;GET POINTER TO LO, ENTRY FROM DINT
        MVI     B,7             ;SET UP A COUNT
DRONA1: INR     M               ;INCREMENT A BYTE
        RNZ                     ;RETURN IF THERE WAS NO CARRY
        INX     H               ;INCREMENT POINTER TO NEXT HIGHER ORDER
        DCR     B               ;HAVE WE INCREMENTED ALL BYTES
        JNZ     DRONA1          ;NO, TRY THE NEXT ONE
        INR     M               ;YES, INCREMENT THE EXPONENT
        JZ      OVERR           ;CHECK FOR OVERFLOW
        DCX     H               ;THE NUMBER OVERFLOWED ITS EXPONENT
        MVI     M,200           ;PUT 200 IN HO
        RET                     ;ALL DONE


        ;ADD OR SUBTRACT 2 DBL QUANTITIES
        ;ALTERS A,C,D,E,H,L
DADDD:  LXI     H,FBUFFR+^D17   ;ENTRY FROM DDIV
        LXI     D,ARGLO         ;ADD OR SUBTRACT FBUFFR+17 AND ARG
```

```
  2097                    34560           JMP     DADDS           ;DO THE OPERATION
  2098                    34580
  2099                    34600   DADDAA: XWD     1000,076        ;"MVI   A", ENTRY FROM DADD, DMULT
  2100                    34620           ADC     M               ;SETUP ADD INSTRUCTION FOR LOOP
  2101                    34640   DADDA:  LXI     H,ARGLO         ;GET POINTER TO ARG, ENTRY FROM DADD
  2102                    34660   DADDFO: LXI     D,DFACLO        ;GET POINTER TO FAC, ENTRY FROM FOUT
  2103                    34680   DADDS:  MVI     C,7             ;SET UP A COUNT
  2104                    34700           STA     DADDOP          ;STORE THE ADD OR SUBTRACT INSTRUCTION
  2105                    34720           XRA     A               ;CLEAR CARRY
  2106                    34740   DADDL:  LDAX    D               ;GET A BYTE FROM RESULT NUMBER
  2107                    34760   DADDOP: NOP                     ;THIS IS EITHER "ADC    M" OR "SBB      M"
  2108                    34780           STAX    D               ;SAVE THE CHANGED BYTE
  2109                    34800           INX     D               ;INCREMENT POINTERS TO NEXT HIGHER ORDER BYTE
  2110                    34820           INX     H
  2111                    34840           DCR     C               ;ARE WE DONE?
  2112                    34860           JNZ     DADDL           ;NO, DO THE NEXT HIGHER ORDER BYTE
  2113                    34880           RET                     ;ALL DONE
  2114                    34900
  2115                    34920
  2116                    34940           ;NEGATE SIGNED NUMBER IN FAC
  2117                    34960           ;THIS IS USED BY DADD, DINT
  2118                    34980           ;ALTERS A,B,C,H,L
  2119                    35000   DNEGR:  MOV     A,M             ;COMPLEMENT SIGN OF FAC
  2120                    35020           CMA                     ;USE THE UNPACKED SIGN BYTE
  2121                    35040           MOV     M,A             ;SAVE THE NEW SIGN
  2122                    35060           LXI     H,DFACLO-1      ;GET POINTER TO LO
  2123                    35080           MVI     B,10            ;SET UP A COUNT
  2124                    35100           XRA     A               ;CLEAR CARRY AND GET A ZERO
  2125                    35120           MOV     C,A             ;SAVE ZERO IN C
  2126                    35140   DNEGR1: MOV     A,C             ;GET A ZERO
  2127                    35160           SBB     M               ;NEGATE THE BYTE OF FAC
  2128                    35180           MOV     M,A             ;UPDATE FAC
  2129                    35200           INX     H               ;INCREMENT POINTER TO NEXT HIGHER ORDER BYTE
  2130                    35220           DCR     B               ;ARE WE DONE?
  2131                    35240           JNZ     DNEGR1          ;NO, NEGATE THE NEXT BYTE
  2132                    35260           RET                     ;ALL DONE
  2133                    35280
  2134                    35300
  2135                    35320           ;SHIFT DBL FAC RIGHT ONE
  2136                    35340           ;A = SHIFT COUNT
  2137                    35360           ;ALTERS A,C,D,E,H,L
  2138                    35380   DSHFTR: LXI     H,DFACLO-1      ;GET POINTER TO LO
  2139                    35400           MVI     M,0             ;PUT ZERO IN EXTRA LO ORDER BYTE
  2140                    35420   DSHFR1: SUI     10              ;SEE IF WE CAN SHIFT 8 RIGHT
  2141                    35440           JC      DSHFR3          ;WE CAN'T, CHECK IF WE ARE DONE
  2142                    35460   DSHFRM: LXI     H,FAC-1         ;ENTRY FROM DMULT, GET POINTER TO HO
  2143                    35480           MVI     E,0             ;SHIFT A ZERO INTO THE HO
  2144                    35500           MVI     D,10            ;SET UP A COUNT
  2145                    35520   DSHFR2: MOV     C,M             ;SAVE A BYTE OF FAC
  2146                    35540           MOV     M,E             ;PUT THE LAST BYTE IN ITS PLACE
  2147                    35560           MOV     E,C             ;SET UP E FOR NEXT TIME THROUGH THE LOOP
  2148                    35580           DCX     H               ;POINT TO NEXT LOWER ORDER BYTE
  2149                    35600           DCR     D               ;ARE WE DONE?
```

```
  2150                    35620           JNZ     DSHFR2          ;NO, DO THE NEXT BYTE
  2151                    35640           JMP     DSHFR1          ;YES, SEE IF WE CAN SHIFT OVER 8 MORE
  2152                    35660   DSHFR3: ADI     11              ;CORRECT SHIFT COUNT
  2153                    35680           MOV     D,A             ;SAVE SHIFT COUNT IN D
  2154                    35700   DSHFR4: XRA     A               ;CLEAR CARRY
  2155                    35720           DCR     D               ;ARE WE DONE?
  2156                    35740           RZ                      ;YES
  2157                    35760   DSHFRA: LXI     H,FAC-1         ;NO, GET POINTER TO LO, ENTRY FROM DADD, DMULT
  2158                    35780           MVI     E,10            ;SET UP A COUNT, ROTATE FAC ONE LEFT
  2159                    35800   DSHFR5: MOV     A,M             ;GET A BYTE OF THE FAC
  2160                    35820           RAR                     ;ROTATE IT LEFT
  2161                    35840           MOV     M,A             ;PUT THE UPDATED BYTE BACK
  2162                    35860           DCX     H               ;DECREMENT POINTER TO NEXT LOWER ORDER BYTE
  2163                    35880           DCR     E               ;ARE WE DONE?
  2164                    35900           JNZ     DSHFR5          ;NO, ROTATE THE NEXT LOWER ORDER BYTE
  2165                    35920           JMP     DSHFR4          ;YES, SEE IF WE ARE DONE SHIFTING
  2166                    35940
  2167                    35960
  2168                    35980           ;ROTATE FAC LEFT ONE
  2169                    36000           ;ALTERS A,C,H,L
  2170                    36020   DSHFLC: MVI     C,10            ;SET UP A COUNT
  2171                    36040   DSHFTL: MOV     A,M             ;GET A BYTE OF FAC
  2172                    36060           RAL                     ;ROTATE IT LEFT ONE
  2173                    36080           MOV     M,A             ;UPDATE BYTE IN FAC
  2174                    36100           INX     H               ;INCREMENT POINTER TO NEXT HIGHER ORDER BYTE
  2175                    36120           DCR     C               ;ARE WE DONE?
  2176                    36140           JNZ     DSHFTL          ;NO, ROTATE THE NEXT BYTE
  2177                    36160           RET                     ;ALL DONE
  2178                    36180
  2179                    36200
  2180                    36220           ;DOUBLE PRECISION MULTIPLICATION        FAC:=ARG*FAC
  2181                    36240           ;ALTERS ALL REGISTERS
  2182                    36260   DMULT:  FSIGN                   ;CHECK IF WE ARE MULTIPLYING BY ZERO
  2183                    36280           RZ                      ;YES, ALL DONE, THE FAC IS ZERO
  2184                    36300           CALL    MULDVA          ;ADD EXPONENTS AND TAKE CARE OF SIGNS
  2185                    36320           CALL    DMULDV          ;ZERO FAC AND PUT FAC IN FBUFFR
  2186                    36340           MOV     M,C             ;PUT UNPACKED HO IN ARG
  2187                    36360           LXI     D,ARGLO         ;GET POINTER TO LO OF ARG
  2188                    36380           MVI     B,7             ;SET UP A COUNT
  2189                    36400   DMULT2: LDAX    D               ;GET THE BYTE OF ARG TO MULTIPLY BY
  2190                    36420           INX     D               ;INCREMENT POINTER TO NEXT HIGHER BYTE
  2191                    36440           ORA     A               ;CHECK IF WE ARE MULTIPLYING BY ZERO
  2192                    36460           PUSH    D               ;SAVE POINTER TO ARG
  2193                    36480           JZ      DMULT5          ;WE ARE
  2194                    36500           MVI     C,10            ;SET UP A COUNT
  2195                    36520   DMULT3: PUSH    B               ;SAVE COUNTERS
  2196                    36540           RAR                     ;ROTATE MULTIPLIER RIGHT
  2197                    36560           MOV     B,A             ;SAVE IT
  2198                    36580           CC      DADDAA          ;ADD IN OLD FAC IF BIT OF MULTIPIER WAS ONE
  2199                    36600           MVI     D,1             ;ROTATE PRODUCT RIGHT ONE
  2200                    36620           CALL    DSHFRA
  2201                    36640           MOV     A,B             ;GET MULTIPLIER IN A
  2202                    36660           POP     B               ;GET COUNTERS BACK
```

```
36680           DCR     C                       ;ARE WE DONE WITH THIS BYTE OF ARG?
36700           JNZ     DMULT3                  ;NO, MULTIPLY BY THE NEXT BIT OF THE MULTIPLIER

36720   DMULT4: POP     D                       ;YES, GET POINTER INTO ARG BACK
36740           DCR     B                       ;ARE WE DONE?
36760           JNZ     DMULT2                  ;NO, MULTIPLY BY NEXT HIGHER ORDER BY OF ARG
36780           JMP     NORMAL                  ;ALL DONE, NORMALIZE AND ROUND RESULT
36800   DMULT5: CALL    DSHFRM                  ;SHIFT PRODUCT RIGHT ONE BYTE, WE ARE
36820           JMP     DMULT4                  ; MULTIPLYIING BY ZERO
36840
36860
36880           ;CONSTANT FOR DIV10, DDIV10
36900   DTEN:   000     ; 10D0
36920           000
36940           000
36960           000
36980   FTEN:   000     ; 10,0
37000           000
37020           040
37040           204
37060
37080           ;DOUBLE PRECISION DIVIDE FAC BY 10
37100           ;ALTERS ALL REGISTERS
37120   DDIV10: CALL    VMOVAF                  ;SAVE THE FAC IN ARG
37140           LXI     H,DTEN                  ;GET POINTER TO A DOUBLE PRECISION 10
37160           CALL    VMOVFM                  ;MOVE TEN INTO THE FAC
37180                                           ;FALL INTO DDIV AND DIVIDE BY TEN
37200
37220
37240           ;DOUBLE PRECISION DIVISION      FAC:=ARG/FAC
37260           ;ALTERS ALL REGISTERS
37280   DDIV:   FSIGN                           ;CHECK FOR DIVISION BY ZERO
37300           JZ      DV0ERR                  ;DON'T LET HIM DO IT
37320           CALL    MULDVS                  ;SUBTRACT EXPONENTS AND CHECK SIGNS
37340           INR     M                       ;ADD TWO TO EXPONENT TO CORRECT SCALING
37360           INR     M
37380           CALL    DMULDV                  ;ZERO FAC AND PUT FAC IN FBUFFR
37400           LXI     H,ARG                   ;GET POINTER TO THE EXTRA HO BYTE WE WILL USE
37420           MOV     M,C                     ;ZERO IT
37440           MVI     B,0                     ;ZERO FLAG TO SEE WHEN WE START DIVIDING
37460   DDIV1:  XWD     1000,076                ;"MVI   A", SUBTRACT FBUFFR FROM ARG
37480           SBB     M                       ;GET SUBTRACT INSTRUCTION
37500           CALL    DADDD                   ;DO THE SUBTRACTION
37520           LDAX    D                       ;SUBTRACT FROM EXTRA HO BYTE
37540           SBB     C                       ;HERE C=0
37560           CMC                             ;CARRY=1 IF SUBTRACTION WAS GOOD
37580           JC      DDIV2                   ;WAS IT OK?
37600           XWD     1000,076                ;"MVI   A"  NO, ADD FBUFFR BACK IN
37620           ADC     M                       ;GET ADD INSTRUCTION
37640           CALL    DADDD                   ;DO THE ADDITION
37660           XRA     A                       ;CLEAR CARRY
37680           XWD     1000,332                ;"JC" OVER NEXT TWO BYTES
37700   DDIV2:  STAX    D                       ;STORE THE NEW HIGHEST ORDER BYTE
```

```
37720           INR     B                       ;INCREMENT FLAG TO SHOW WE COULD DIVIDE
37740           LDA     FAC-1                   ;CHECK IF WE ARE DONE DIVIDING
37760           INR     A                       ;SET SIGN FLAG WITHOUT AFFECTING CARRY
37780           DCR     A
37800           RAR                             ;PUT CARRY IN MSB FOR DROUND
37820           JM      DROUNB                  ;WE ARE DONE, WE HAVE 57 BITS OF ACCURACY
37840           RAL                             ;GET OLD CARRY BACK WHERE IT BELONGS
37860           LXI     H,DFACLO                ;GET POINTER TO LO OF FAC
37880           MVI     C,7                     ;SET UP A COUNT, SHIFT FAC LEFT ONE
37900           CALL    DSHFTL                  ;SHIFT IN THE NEXT BIT IN THE QUOTIENT
37920           LXI     H,ARGLO                 ;GET POINTER TO LO IN ARG
37940           CALL    DSHFLC                  ;SHIFT DIVIDEND ONE LEFT
37960           MOV     A,B                     ;IS THIS THE FIRST TIME AND WAS THE
37980           ORA     A                       ; SUBTRACTION NOT GOOD? (B WILL GET
38000           JNZ     DDIV1                   ; CHANGED ON THE FIRST OR SECOND SUBTRACTION)
38020           LXI     H,FAC                   ;YES, SUBTRACT ONE FROM EXPONENT TO CORRECT
38040           DCR     M                       ; SCALING
38060           JNZ     DDIV1                   ;CONTINUE DIVIDING IF NO OVERFLOW
38080           JMP     OVERR                   ;WE HAVE OVERFLOW!!
38100
38120
38140           ;TRANSFER FAC TO FBUFFR FOR DMULT AND DDIV
38160           ;ALTERS A,B,C,D,E,H,L
38180   DMULDV: MOV     A,C                     ;PUT UNPACKED HO BACK IN ARG
38200           STA     ARG-1
38220           DCX     H                       ;POINT TO HO OF FAC
38240           LXI     D,FBUFFR+^D23           ;POINT TO END OF FBUFFR
38260           MVI     B,7                     ;SET UP A COUNT
38280           MVI     C,0                     ;GET A ZERO TO FILL FAC WITH
38300   DMLDV1: MOV     A,M                     ;GET A BYTE FROM FAC
38320           STAX    D                       ;PUT IT IN FBUFFR
38340           MOV     M,C                     ;PUT A ZERO IN FAC
38360           DCX     D                       ;POINT TO NEXT BYTE IN FBUFFR
38380           DCX     H                       ;POINT TO NEXT LOWER ORDER BYTE IN FAC
38400           DCR     B                       ;ARE WE DONE?
38420           JNZ     DMLDV1                  ;NO, TRANSFER THE NEXT BYTE
38440           RET                             ;ALL DONE
38460
38480
38500           ;DOUBLE PRECISION MULTIPLY THE FAC BY 10
38520           ;ALTERS ALL REGISTERS
38540   DMUL10: CALL    VMOVAF                  ;SAVE THE FAC IN ARG
38560                                           ;VMOVAF EXITS WITH (DE)=FAC+1
38580           XCHG                            ;GET THE POINTER INTO THE FAC IN (HL)
38600           DCX     H                       ;POINT TO THE EXPONENT
38620           MOV     A,M                     ;GET THE EXPONENT
38640           ADI     2                       ;MULTIPLY FAC BY 4 BY ADDING 2 TO THE EXPONENT
38660           JC      OVERR                   ;CHECK FOR OVERFLOW
38680           MOV     M,A                     ;SAVE THE NEW EXPONENT
38700           PUSH    H                       ;SAVE POINTER TO FAC
38720           CALL    DADD                    ;ADD IN THE ORIGINAL FAC TO GET 5 TIMES  FAC
38740           POP     H                       ;GET THE POINTER TO FAC BACK
38760           INR     M                       ;ADD ONE TO EXPONENT TO GET 10 TIMES FAC
```