```
                                SUBTTL  GREATEST INTEGER FUNCTION
                                        ;QUICK GREATEST INTEGER FUNCTION
                                        ;LEAVES INT(FAC) IN C,D,E (SIGNED)
                                        ;ASSUMES FAC .LT. 2^23 = 8388608
                                        ;ASSUMES THE EXPONENT OF FAC IS IN A
                                        ;ALTERS A,B,C,D,E
001372' 001000  000107          QINT:   MOV     B,A             ;ZERO B,C,D,E IN CASE THE NUMBER IS ZERO
001373' 001000  000117                  MOV     C,A
001374' 001000  000127                  MOV     D,A
001375' 001000  000137                  MOV     E,A
001376' 001000  000267                  ORA     A               ;SET CONDITION CODES
001377' 001000  000310                  RZ                      ;IT IS ZERO, WE ARE DONE

                                ;THE HARD CASE IN QINT IS NEGATIVE NON-INTEGERS.  TO HANDLE THIS, IF THE
                                ;NUMBER IS NEGATIVE, WE REGARD THE 3-BYTE MANTISSA AS A 3-BYTE INTEGER AND
                                ;SUBTARCT ONE.  THEN ALL THE FRACTIONAL BITS ARE SHIFTED OUT BY SHIFTING THE
                                ;MANTISSA RIGHT.  THEN, IF THE NUMBER WAS NEGATIVE, WE ADD ONE.  SO, IF WE
                                ;HAD A NEGATIVE INTEGER, ALL THE BITS TO THE RIGHT OF THE BINARY POINT WERE
                                ;ZERO.  SO THE NET EFFECT IS WE HAVE THE ORIGINAL NUMBER IN C,D,E.  IF THE
                                ;NUMBER WAS A NEGATIVE NON-INTEGER, THERE IS AT LEAST ONE NON-ZERO BIT TO THE
                                ;RIGHT OF THE BINARY POINT.  SO THE NET EFFECT IS THAT WE GET THE ABSOLUTE
                                ;VALUE OF INT(FAC) IN C,D,E.  C,D,E IS THEN NEGATED IF THE ORIGINAL NUMBER WAS
                                ;NEGATIVE SO THE RESULT WILL BE SIGNED.
001400' 001000  000345                  PUSH    H               ;SAVE (HL)
001401' 001000  000315                  CALL    MOVRF           ;GET NUMBER IN THE REGISTERS
001402' 000000  001240'
001403' 000000  001342'
001404' 001000  000315                  CALL    UNPACK          ;UNPACK THE NUMBER
001405' 000000  001272'
001406' 000000  001402'
001407' 001000  000256                  XRA     M               ;GET SIGN OF NUMBER
001410' 001000  000147                  MOV     H,A             ;DON'T LOSE IT
001411' 001000  000374                  CM      QINTA           ;SUBTRACT 1 FROM LO IF NUMBER IS NEGATIVE
001412' 000000  001436'
001413' 000000  001405'
001414' 001000  000076                  MVI     A,230           ;SEE HOW MANY WE HAVE TO SHIFT TO CHANGE
001415' 000000  000230
001416' 001000  000220                  SUB     B               ; NUMBER TO AN INTEGER
001417' 001000  000315                  CALL    SHIFTR          ;SHIFT NUMBER TO GET RID OF FRACTIONAL BITS
001420' 000000  000334'
001421' 000000  001412'
001422' 001000  000174                  MOV     A,H             ;GET SIGN
001423' 001000  000027                  RAL                     ;PUT SIGN IN CARRY SO IT WILL NOT BE CHANGED
001424' 001000  000334                  CC      ROUNDA          ;IF NUMBER WAS NEGATIVE, ADD ONE
001425' 000000  000255'
001426' 000000  001420'
001427' 001000  000006                  MVI     B,0             ;FORGET THE BITS WE SHIFTED OUT
001430' 000000  000000
001431' 001000  000334                  CC      NEGR            ;NEGATE NUMBER IF IT WAS NEGATIVE BECAUSE WE
001432' 000000  000310'
001433' 000000  001425'
                                                                ; WANT A SIGNED MANTISSA
001434' 001000  000341                  POP     H               ;GET OLD (HL) BACK
```

```
001435' 001000  000311                  RET                     ;ALL DONE

001436' 001000  000033          QINTA:  DCX     D               ;SUBTRACT ONE FROM C,D,E
001437' 001000  000172                  MOV     A,D             ;WE HAVE TO SUBTRACT ONE FROM C IF
001440' 001000  000243                  ANA     E               ; D AND E ARE BOTH ALL ONES
001441' 001000  000074                  INR     A               ;SEE IF BOTH WERE -1
001442' 001000  000300                  RNZ                     ;THEY WERE NOT, WE ARE DONE
                                IFN     LENGTH-2,<
001443' 001000  000015                  DCR     C>              ;THEY WERE, SUBTRACT ONE FROM C
                                IFE     LENGTH-2,<
                                DCXBRT: DCX     B>              ;THIS IS FOR BILL.  C WILL NEVER BE ZERO
                                                                ; (THE MSB WILL ALWAYS BE ONE) SO "DCX  B"
                                                                ; AND "DCR     C" ARE FUNCTIONALLY EQUIVALENT
001444' 001000  000311                  RET                     ;ALL DONE


                                        ;GREATEST INTEGER FUNCTION
                                        ;ALTERS A,B,C,D,E,H,L
                                IFE     LENGTH-2,<
                                INTFNC: CPI     4               ;SEE WHAT KIND OF NUMBER WE HAVE
                                        RC                      ;IT IS AN INTEGER, ALL DONE
                                        JNZ     DINT            ;CONVERT THE DOUBLE PRECISION NUMBER
                                        CALL    CONIS>          ;TRY TO CONVERT THE NUMBER TO AN INTEGER
                                                                ;IF WE CAN'T, WE WILL RETURN HERE TO GIVE A
                                                                ; SINGLE PRECISION RESULT
001445' 001000  000041          INT:    LXI     H,FAC           ;GET EXPONENT
001446' 000000  001156*
001447' 000000  001432'
001450' 001000  000176                  MOV     A,M
001451' 001000  000376                  CPI     230             ;SEE IF NUMBER HAS ANY FRACTIONAL BITS
001452' 000000  000230
                                IFN     EXTFNC,<                ;THE ONLY GUY WHO NEEDS THIS DOESN'T CARE
001453' 001000  000072                  LDA     FACLO>          ; ABOUT THE SIGN
001454' 000000  001255*
001455' 000000  001446'
001456' 001000  000320                  RNC                     ;IT DOES NOT
                                IFN     EXTFNC,<
001457' 001000  000176                  MOV     A,M>            ;GET EXPONENT BACK
001460' 001000  000315                  CALL    QINT            ;IT DOES, SHIFT THEM OUT
001461' 000000  001372'
001462' 000000  001454'
001463' 001000  000066                  MVI     M,230           ;CHANGE EXPONENT SO IT WILL BE CORRECT
001464' 000000  000230
                                                                ; AFTER NORMALIZATION
                                IFN     EXTFNC,<
001465' 001000  000173                  MOV     A,E             ;GET LO
001466' 001000  000365                  PUSH    PSW>            ;SAVE IT
001467' 001000  000171                  MOV     A,C             ;NEGATE NUMBER IF IT IS NEGATIVE
001470' 001000  000027                  RAL                     ;PUT SIGN IN CARRY
                                IFE     EXTFNC,<
                                        JMP     FADFLT>         ;REFLOAT NUMBER
                                IFN     EXTFNC,<
001471' 001000  000315                  CALL    FADFLT          ;REFLOAT NUMBER
```

```
001472' 000000  000143'
001473' 000000  001461'
001474' 001000  000361          25500   POPPRT: POP     PSW             ;GET LO BACK
001475' 001000  000311          25520           RET>                    ;ALL DONE


                                25580   IFE     LENGTH-2,<
                                25600           ;GREATEST INTEGER FUNCTION FOR DOUBLE PRECISION NUMBERS
                                25620           ;ALTERS A,B,C,D,E,H,L
                                25640   DINT:   LXI     H,FAC           ;GET POINTER TO FAC
                                25660           MOV     A,M             ;GET EXPONENT
                                25680           CPI     220             ;CAN WE CONVERT IT TO AN INTEGER?
                                25700           JC      FRCINT          ;THEN DO SO
                                25720           JNZ     DINT2           ;CHECK FOR -32768
                                25740           MOV     C,A             ;SAVE EXPONENT IN C
                                25760           DCX     H               ;GET POINTER TO SIGN AND HO
                                25780           MOV     A,M             ;GET SIGN AND HO
                                25800           XRI     200             ;CHECK IF IT IS 200
                                25820           MVI     B,6             ;SET UP A COUNT TO CHECK IF THE REST OF
                                25840   DINT1:  DCX     H               ; THE NUMBER IS ZERO, POINT TO NEXT BYTE
                                25860           ORA     M               ;IF ANY BITS ARE NON-ZERO, A WILL BE NON-ZERO
                                25880           DCR     B               ;ARE WE DONE?
                                25900           JNZ     DINT1           ;NO, CHECK THE NEXT LOWER ORDER BYTE
                                25920           ORA     A               ;IS A NOW ZERO?
                                25940           LXI     H,200*400+$CODE ;GET -32768 JUST IN CASE
                                25960           JZ      CONISS          ;A IS ZERO SO WE HAVE -32768
                                25980           MOV     A,C             ;GET EXPONENT
                                26000   DINT2:  CPI     270             ;ARE THERE ANY FRACTIONAL BITS?
                                26020           RNC                     ;NO, THE NUMBER IS ALREADY AN INTEGER
                                26040   DINTFO: PUSH    PSW             ;ENTRY FROM FOUT, CARRY IS ZERO IF WE COME
                                26060                                   ; HERE FROM FOUT
                                26080           CALL    MOVRF           ;GET HO'S OF NUMBER IN REGISTERS FOR UNPACKING
                                26100           CALL    UNPACK          ;UNPACK IT
                                26120           XRA     M               ;GET ITS SIGN BACK
                                26140           DCX     H               ;SET THE EXPONENT TO NORMALIZE CORRECTLY
                                26160           MVI     M,270
                                26180           PUSH    PSW             ;SAVE THE SIGN
                                26200           CM      DINTA           ;SUBTRACT 1 FROM LO IF NUMBER IS NEGATIVE
                                26220           MVI     A,270           ;GET HOW MANY BITS WE HAVE TO SHIFT OUT
                                26240           SUB     B
                                26260           CALL    DSHFTR          ;SHIFT THEM OUT!!
                                26280           POP     PSW             ;GET THE SIGN BACK
                                26300           CM      DROUNA          ;IF NUMBER WAS NEGATIVE, ADD ONE
                                26320           XRA     A               ;PUT A ZERO IN THE EXTRA LO BYTE SO WHEN
                                26340           STA     DFACLO-1        ; WE NORMALIZE, WE WILL SHIFT IN ZEROS
                                26360           POP     PSW             ;IF WE WERE CALLED FROM FOUT, DON'T NORMALIZE,
                                26380           RNC                     ; JUST RETURN
                                26400           JMP     DNORML          ;RE-FLOAT THE INTEGER
                                26420
                                26440   DINTA:  LXI     H,DFACLO        ;SUBTRACT ONE FROM FAC, GET POINTER TO LO
                                26460   DINTA1: MOV     A,M             ;GET A BYTE OF FAC
                                26480           DCR     M               ;SUBTRACT ONE FROM IT
                                26500           ORA     A               ;CONTINUE ONLY IF THE BYTE USED TO BE ZERO
```

```
                                26520           INX     H               ;INCREMENT POINTER TO NEXT BYTE
                                26540           JZ      DINTA1          ;CONTINUE IF NECESSARY
                                26560           RET>                    ;ALL DONE
                                26580   PAGE
                                26600   SUBTTL  INTEGER ARITHMETIC ROUTINES
                                26620   IFN     MULDIM&<LENGTH-2>,<
                                26640           ;TWO BYTE UNSIGNED INTEGER MULTIPLY
                                26660           ; (HL):=(BC)*(DE)
                                26680           ;A,D,E,H,L ARE CHANGED
001476' 001000  000041          26700   DMULT:  LXI     H,$CODE         ;ZERO PRODUCT REGISTERS
001477' 000000  001153*
001500' 000000  001472'
001501' 001000  000170          26720           MOV     A,B             ;CHECK IF (BC) IS ZERO
001502' 001000  000261          26740           ORA     C               ;IF SO, JUST RETURN, (HL) IS ALREADY ZERO
001503' 001000  000310          26760           RZ                      ;THIS IS DONE FOR SPEED
001504' 001000  000076          26780           MVI     A,20            ;SET UP A COUNT
001505' 000000  000020
001506' 001000  000051          26800   DMULT1: DAD     H               ;ROTATE (HL) LEFT ONE
001507' 001000  000332          26820           JC      BSERR##         ;CHECK FOR OVERFLOW, IF SO,
001510' 000000  000000*
001511' 000000  001477'
001512' 001000  000353          26840           XCHG                    ; BAD SUBSCRIPT (BS) ERROR
001513' 001000  000051          26860           DAD     H               ;ROTATE (DE) LEFT ONE
001514' 001000  000353          26880           XCHG
001515' 001000  000322          26900           JNC     DMULT2          ;ADD IN (BC) IF HO WAS 1
001516' 000000  001524'
001517' 000000  001510'
001520' 001000  000011          26920           DAD     B
001521' 001000  000332          26940           JC      BSERR           ;CHECK FOR OVERFLOW
001522' 000000  001510*
001523' 000000  001516'
001524' 001000  000075          26960   DMULT2: DCR     A               ;SEE IF DONE
001525' 001000  000302          26980           JNZ     DMULT1
001526' 000000  001506'
001527' 000000  001522'
001530' 001000  000311          27000           RET>                    ;ALL DONE


                                27060   IFE     LENGTH-2,<
                                27080   COMMENT %
                                27100           INTEGER ARITHMETIC CONVENTIONS
                                27120
                                27140   INTEGER VARIABLES ARE 2 BYTE, SIGNED NUMBERS
                                27160           THE LO BYTE COMES FIRST IN MEMORY
                                27180
                                27200   CALLING CONVENTIONS:
                                27220   FOR ONE ARGUMENT FUNCTIONS:
                                27240           THE ARGUMENT IS IN (HL), THE RESULT IS LEFT IN (HL)
                                27260   FOR TWO ARGUMENT OPERATIONS:
                                27280           THE FIRST ARGUMENT IS IN (DE)
                                27300           THE SECOND ARGUMENT IS IN (HL)
                                27320           THE RESULT IS LEFT IN (HL)
                                27340   IF OVERFLOW OCCURS, THE ARGUMENTS ARE CONVERTED TO SINGLE PRECISION
                                27360   WHEN INTEGERS ARE STORED IN THE FAC, THEY ARE STORED AT FACLO+0,1
                                27380   VALTYP(INTEGER)=2
                                27400   %
                                27420
```

```
                                27440
                                27460           ;INTEGER SUBTRTACTION   (HL):=(DE)-(HL)
                                27480           ;ALTERS A,B,C,D,E,H,L
                                27500   ISUB:   MOV     A,H             ;EXTEND THE SIGN OF (HL) TO B
                                27520           RAL                     ;GET SIGN IN CARRY
                                27540           SBB     A
                                27560           MOV     B,A
                                27580           CALL    INEGHL          ;NEGATE (HL)
                                27600           MOV     A,C             ;GET A ZERO
                                27620           SBB     B               ;NEGATE SIGN
                                27640           JMP     IADDS           ;GO ADD THE NUMBERS
                                27660
                                27680
                                27700           ;INTEGER ADDITION       (HL):=(DE)+(HL)
                                27720           ;ALTERS A,B,C,D,E,H,L
                                27740   IADD:   MOV     A,H             ;EXTEND THE SIGN OF (HL) TO B
                                27760           RAL                     ;GET SIGN IN CARRY
                                27780           SBB     A
                                27800   IADDS:  MOV     B,A             ;SAVE THE SIGN
                                27820           PUSH    H               ;SAVE THE SECOND ARGUMENT IN CASE OF OVERFLOW
                                27840           MOV     A,D             ;EXTEND THE SIGN OF (DE) TO A
                                27860           RAL                     ;GET SIGN IN CARRY
                                27880           SBB     A
                                27900           DAD     D               ;ADD THE TWO LO'S
                                27920           ADC     B               ;ADD THE EXTRA HO
                                27940           RRC                     ;IF THE LSB OF A IS DIFFERENT FROM THE MSB OF
                                27960           XRA     H               ; H, THEN OVERFLOW OCCURED
                                27980           JP      POPPRT          ;NO OVERFLOW, GET OLD (HL) OFF STACK AND WE
                                28000                                   ; ARE DONE
                                28020           PUSH    B               ;OVERFLOW -- SAVE EXTENDED SIGN OF (HL)
                                28040           XCHG                    ;GET (DE) IN (HL)
                                28060           CALL    CONSIH          ;FLOAT IT
                                28080           POP     PSW             ;GET SIGN OF (HL) IN A
                                28100           POP     H               ;GET OLD (HL) BACK
                                28120           CALL    PUSHF           ;PUT FIRST ARGUMENT ON STACK
                                28140           XCHG                    ;PUT SECOND ARGUMENT IN (DE) FOR FLOATR
                                28160           CALL    INEGAD          ;FLOAT IT
                                28180           POPR                    ;GET FIRST ARGUMENT OFF STACK
                                28200           JMP     FADD            ;ADD THE TWO NUMBERS USING SINGLE PRECISION
                                28220
                                28240
                                28260           ;INTEGER MULTIPLICATION         (HL):=(DE)*(HL)
                                28280           ;ALTERS A,B,C,D,E,H,L
                                28300   IMULT:  PUSH    H               ;SAVE SECOND ARGUMENT IN CASE OF OVERFLOW
                                28320           PUSH    D               ;SAVE FIRST ARGUMENT
                                28340           CALL    IMULDV          ;FIX UP THE SIGNS
                                28360           PUSH    B               ;SAVE THE SIGN OF THE RESULT
                                28380           MOV     B,H             ;COPY SECOND ARGUMENT INTO (BC)
                                28400           MOV     C,L
                                28420           LXI     H,$CODE         ;ZERO (HL), THAT IS WHERE THE PRODUCT GOES
                                28440           MVI     A,20            ;SET UP A COUNT
                                28460   IMULT1: DAD     H               ;ROTATE PRODUCT LEFT ONE
                                28480           JC      IMULT5          ;CHECK FOR OVERLFOW
```

```
28500           XCHG                    ;ROTATE FIRST ARGUMENT LEFT ONE TO SEE IF
28520           DAD     H               ; WE ADD IN (BC) OR NOT
28540           XCHG
28560           JNC     IMULT2          ;DON'T ADD IN ANYTHING
28580           DAD     B               ;ADD IN (BC)
28600           JC      IMULT5          ;CHECK FOR OVERLFOW
28620   IMULT2: DCR     A               ;ARE WE DONE?
28640           JNZ     IMULT1          ;NO, DO IT AGAIN
28660           POP     B               ;WE ARE DONE, GET SIGN OF RESULT
28680           POP     D               ;GET ORIGINAL FIRST ARGUMENT
28700   IMLDIV: MOV     A,H             ;ENTRY FROM IDIV, IS RESULT .GE. 32768?
28720           ORA     A
28740           JM      IMULT3          ;IT IS, CHECK FOR SPECIAL CASE OF -32768
28760           POP     D               ;RESULT IS OK, GET SECOND ARGUMENT OFF STACK
28780           MOV     A,B             ;GET THE SIGN OF RESULT IN A
28800           JMP     INEGA           ;NEGATE THE RESULT IF NECESSARY
28820   IMULT3: XRI     200             ;IS RESULT 32768?
28840           ORA     L               ;NOTE: IF WE GET HERE FROM IDIV, THE RESULT
28860           JZ      IMULT4          ; MUST BE 32768, IT CANNOT BE GREATER
28880           XCHG                    ;IT IS .GT. 32768, WE HAVE OVERFLOW
28900           XWD     1000,001        ;"LXI   B" OVER NEXT 2 BYTES
28920   IMULT5: POP     B               ;GET SIGN OF RESULT OFF STACK
28940           POP     H               ;GET THE ORIGINAL FIRST ARGUMENT
28960           CALL    CONSIH          ;FLOAT IT
28980           POP     H               ;GET THE ORIGINAL SECOND ARGUMENT
29000           CALL    PUSHF           ;SAVE FLOATED FIRST ARUMENT
29020           CALL    CONSIH          ;FLOAT SECOND ARGUMENT
29040   FMULTT: POPR                    ;GET FIRST ARGUMENT OFF STACK, ENTRY FROM POLYX

29060           JMP     FMULT           ;MULTIPLY THE ARGUMENTS USING SINGLE PRECISION
29080   IMULT4: MOV     A,B             ;IS RESULT +32768 OR -32768?
29100           ORA     A               ;GET ITS SIGN
29120           POP     B               ;DISCARD ORIGINAL SECOND ARGUMENT
29140           RM                      ;THE RESULT SHOULD BE NEGATIVE, IT IS OK
29160           PUSH    D               ;IT IS POSITIVE, SAVE REMAINDER FOR MOD
29180           CALL    CONSIH          ;FLOAT -32768
29200           POP     D               ;GET MOD'S REMAINDER BACK
29220           JMP     NEG             ;NEGATE -32768 TO GET 32768, WE ARE DONE
29240
29260
29280           ;INTEGER DIVISION       (HL):=(DE)/(HL)
29300           ;REMAINDER IS IN (DE), QUOTIENT IN (HL)
29320           ;ALTERS A,B,C,D,E,H,L
29340   IDIV:   MOV     A,H             ;CHECK FOR DIVISION BY ZERO
29360           ORA     L
29380           JZ      DV0ERR          ;WE HAVE DIVISION BY ZERO!!
29400           CALL    IMULDV          ;FIX UP THE SIGNS
29420           PUSH    B               ;SAVE THE SIGN OF THE RESULT
29440           XCHG                    ;GET DENOMINATOR IN (HL)
29460           CALL    INEGHL          ;NEGATE IT
29480           MOV     B,H             ;SAVE NEGATED DENOMINATOR IN (BC)
29500           MOV     C,L
29520           LXI     H,$CODE         ;ZERO WHERE WE DO THE SUBTRACTION
```

```
29540           MVI     A,21            ;SET UP A COUNT
29560           PUSH    PSW             ;SAVE IT
29580           ORA     A               ;CLEAR CARRY
29600           JMP     IDIV3           ;GO DIVIDE
29620   IDIV1:  PUSH    PSW             ;SAVE COUNT
29640           PUSH    H               ;SAVE (HL) I.E. CURRENT NUMERATOR
29660           DAD     B               ;SUBTRACT DENOMINATOR
29680           JNC     IDIV2           ;WE SUBTRACTED TOO MUCH, GET OLD (HL) BACK
29700           POP     PSW             ;THE SUBTRACTION WAS GOOD, DISCARD OLD (HL)
29720           STC                     ;NEXT BIT IN QUOTIENT IS A ONE
29740           XWD     1000,076        ;"MVI   A" OVER NEXT BYTE
29760   IDIV2:  POP     H               ;IGNORE THE SUBTRACTION, WE COULDN'T DO IT
29780   IDIV3:  MOV     A,E             ;SHIFT IN THE NEXT QUOTIENT BIT
29800           RAL
29820           MOV     E,A
29840           MOV     A,D             ;SHIFT THE HO
29860           RAL
29880           MOV     D,A
29900           MOV     A,L             ;SHIFT IN THE NEXT BIT OF THE NUMERATOR
29920           RAL
29940           MOV     L,A
29960           MOV     A,H             ;DO THE HO
29980           RAL
30000           MOV     H,A             ;SAVE THE HO
30020           POP     PSW             ;GET COUNT BACK
30040           DCR     A               ;ARE WE DONE?
30060           JNZ     IDIV1           ;NO, DIVIDE AGAIN
30080           XCHG                    ;GET QUOTIENT IN (HL), REMAINDER IN (DE)
30100           POP     B               ;GET SIGN OF RESULT
30120           PUSH    D               ;SAVE REMAINDER SO STACK WILL BE ALRIGHT
30140           JMP     IMLDIV          ;CHECK FOR SPECIAL CASE OF 32768
30160
30180
30200           ;GET READY TO MULTIPLY OR DIVIDE
30220           ;ALTERS A,B,C,D,E,H,L
30240   IMULDV: MOV     A,H             ;GET SIGN OF RESULT
30260           XRA     D
30280           MOV     B,A             ;SAVE IT IN B
30300           CALL    INEGH           ;NEGATE SECOND ARGUMENT IF NECESARY
30320           XCHG                    ;PUT (DE) IN (HL), FALL IN AND NEGATE FIRST
30340                                   ; ARGUMENT IF NECESSARY
30360
30380
30400           ;NEGATE H,L
30420           ;ALTERS A,C,H,L
30440   INEGH:  MOV     A,H             ;GET SIGN OF (HL)
30460   INEGA:  ORA     A               ;SET CONDITION CODES
30480           RP                      ;WE DON'T HAVE TO NEGATE, IT IS POSITIVE
30500   INEGHL: XRA     A               ;CLEAR A
30520           MOV     C,A             ;STORE A ZERO (WE USE THIS METHOD FOR ISUB)
30540           SUB     L               ;NEGATE LO
30560           MOV     L,A             ;SAVE IT
30580           MOV     A,C             ;GET A ZERO BACK
```