```
                                        45200           ;HERE WE PACK IN THE NEXT DIGIT OF THE EXPONENT
                                        45220           ;WE MUTIPLY THE OLD EXPONENT BY TEN AND ADD IN THE NEXT DIGIT
                                        45240           ;NOTE: EXPONENT OVERFLOW IS NOT CHECKED FOR
001742' 001000  000173                  45260   FINEDG: MOV     A,E             ;EXPONENT DIGIT -- MULTIPLY EXPONENT BY 10
001743' 001000  000007                  45280           RLC                     ;FIRST BY 4
001744' 001000  000007                  45300           RLC
001745' 001000  000203                  45320           ADD     E               ;ADD 1 TO MAKE 5
001746' 001000  000007                  45340           RLC                     ;NOW DOUBLE TO GET 10
001747' 001000  000206                  45360           ADD     M               ;ADD IT IN
001750' 001000  000326                  45380           SUI     "0"             ;SUBTRACT OFF ASCII CODE
001751' 000000  000060
001752' 001000  000137                  45400           MOV     E,A             ;STORE EXPONENT
001753' 001000  000303                  45420           JMP     FINEC           ;CONTINUE
001754' 000000  001621'
001755' 000000  001740'
                                        45440   PAGE
```



```
                                        45460   SUBTTL  FLOATING POINT OUTPUT ROUTINE
                                        45480           ;ENTRY TO LINPRT
001756' 001000  000345                  45500   INPRT:  PUSH    H               ;SAVE LINE NUMBER
001757' 001000  000041                  45520           LXI     H,INTXT##       ;PRINT MESSAGE
001760' 000000  000000*
001761' 000000  001754'
001762' 001000  000315                  45540           CALL    STROUT
001763' 000000  000000*
001764' 000000  001760'
001765' 001000  000341                  45560           POP     H               ;FALL INTO LINPRT

                                        45620           ;PRINT THE 2 BYTE NUMBER IN H,L
                                        45640           ;ALTERS ALL REGISTERS
001766'                                 45660   LINPRT:
                                        45680   IFN     LENGTH-2,<
001766' 001000  000353                  45700           XCHG                    ;SET UP REGISTERS FOR FLOATR
001767' 001000  000257                  45720           XRA     A
001770' 001000  000006                  45740           MVI     B,230
001771' 000000  000230
001772' 001000  000315                  45760           CALL    FLOATR#         ;CONVERT TO FLOATING POINT
001773' 000000  001155'
001774' 000000  001763'
                                        45780   IFE     LENGTH-2,<
                                        45800           CALL    CONISS          ;PUT THE LINE NUMBER IN THE FAC AS AN INTEGER
                                        45820           XRA     A               ;SET FORMAT TO FREE FORMAT
                                        45840           CALL    FOUINI#         ;SET UP THE SIGN
001775' 001000  000041                  45860           LXI     H,STROUI##      ;PUT PRINT STRING ADDRESS ON STACK SO WE WILL
001776' 000000  000000*
001777' 000000  001773'
002000' 001000  000345                  45880           PUSH    H               ; RETURN TO IT AND DO AN "INX   H"
                                        45900           ;THIS GETS RID OF THE SPACE FOR THE SIGN AT
                                        45920           ; THE BEGINNING OF A LINE NUMBER
                                        45940           ;FALL INTO FOUT
                                        45960   IFE     LENGTH-2,<
                                        45980           PUSH    B               ;PUT DUMMY FIELD LENGTHS ON STACK
                                        46000           JMP     FOUT2           ;PRINT THE NUMBER

                                        46060           ;FLOATING OUTPUT OF FAC
                                        46080           ;ALTERS ALL REGISTERS
                                        46100           ;THE ORIGINAL CONTENTS OF THE FAC IS LOST
                                        46120   IFN     LENGTH-2,<
002001' 001000  000041                  46140   FOUT:   LXI     H,FBUFFR        ;GET BEGINING OF CHARACTER BUFFER
002002' 000000  000000*
002003' 000000  001776'
002004' 001000  000345                  46160           PUSH    H               ;SAVE IT FOR WHEN WE RETURN
                                        46180           ;PUT THE SIGN OF THE NUMBER IN THE BUFFER AND MAKE IT POSITIVE
002005' 001000  000357                  46200           FSIGN                   ;GET SIGN OF NUMBER
002006' 001000  000066                  46220           MVI     M," "           ;PRINT SPACE IF POSITIVE
002007' 000000  000040
002010' 001000  000362                  46240           JP      FOUT1
002011' 000000  002015'
```

```
002012' 000000  002002'
002013' 001000  000066     46260           MVI     M,"-"           ;PRINT A MINUS SIGN IF NEGATIVE
002014' 000000  000055
002015' 001000  000043     46280   FOUT1:  INX     H               ;INCREMENT POINTER TO NEXT CHRACTER POSITION
002016' 001000  000066     46300           MVI     M,"0"           ;PUT A ZERO IN BUFFER IN CASE NUMBER=0
002017' 000000  000060
002020' 001000  000312     46320           JZ      FOUT19          ;DO IT IF THE NUMBER IS ZERO
002021' 000000  002266'
002022' 000000  002011'
002023' 001000  000345     46340           PUSH    H               ;SAVE BUFFER POINTER
002024' 001000  000374     46360           CM      NEG             ;NEGATE NUMBER IF NEGATIVE
002025' 000000  001175'
002026' 000000  002021'

                           46400   ;HERE WE GET THE FAC IN THE RANGE 100000 .LE. FAC .LE. 999999 AND ROUND IT TO
                           46420   ;AN INTEGER.  WE KEEP A COUNT OF HOW MANY TIMES WE MULTIPLY OR DIVIDE BY TEN
                           46440   ;SO WE KNOW WHAT THE EXPONENT WILL BE.  THE FAC IS THEN CONVERTED TO AN
                           46460   ;INTEGER IN C,D,E.  WE USE A TABLE OF POWERS OF TEN TO CALCULATE EACH DIGIT.
                           46480   ;THIS ALGORITHM IS USED FOR SPEED.
002027' 001000  000257     46500           XRA     A               ;PUT TEN'S EXPONENT COUNT ON STACK
002030' 001000  000365     46520           PUSH    PSW
002031' 001000  000315     46540           CALL    FOUTCB          ;SEE IF NUMBER IS TOO BIG OR TOO SMALL
002032' 000000  002274'
002033' 000000  002025'
002034' 001000  000001     46560   FOUT3:  MOVRI   221,103,117,370 ;IS NUMBER .LE. 99999.9499?  IT IS TOO SMALL
002035' 000000  000103
002036' 000000  000221
002037' 001000  000021
002040' 000000  000370
002041' 000000  000117
002042' 001000  000315     46580           CALL    FCOMP           ;FCOMP RETURNS 377, 0 OR 1 IN A, SO THE
002043' 000000  001317'
002044' 000000  002032'
                           46600                                   ; PARITY WILL BE ODD IFF 1 IS RETURNED
002045' 001000  000342     46620           JPO     FOUT5           ;NO, NUMBER IS IN RANGE
002046' 000000  002071'
002047' 000000  002043'
002050' 001000  000361     46640           POP     PSW
002051' 001000  000315     46660           CALL    FINMLT          ;YES, MULTIPLY IT BY TEN TO GET
002052' 000000  001673'
002053' 000000  002046'
002054' 001000  000365     46680           PUSH    PSW             ; IT IN RANGE
002055' 001000  000303     46700           JMP     FOUT3           ;SEE IF NUMBER IS NOW IN RANGE
002056' 000000  002034'
002057' 000000  002052'
002060' 001000  000315     46720   FOUT9:  CALL    DIV10           ;NO, DIVIDE NUMBER BY TEN, IT IS TOO BIG
002061' 000000  000637'
002062' 000000  002056'
002063' 001000  000361     46740           POP     PSW             ;ADD ONE TO EXPONENT
002064' 001000  000074     46760           INR     A
002065' 001000  000365     46780           PUSH    PSW
002066' 001000  000315     46800           CALL    FOUTCB          ;IS NUMBER .LE. 999999.499?
002067' 000000  002274'
```



```
002070' 000000  002061'
                           46820                                   ;YES, NUMBER IS IN PRINTING RANGE, I.E.
                           46840                                   ; ALL DIGITS TO BE PRINTED ARE THE INTEGER PART

002071' 001000  000315     46860   FOUT5:  CALL    FADDH           ;ROUND NUMBER TO NEAREST INTEGER
002072' 000000  000000'
002073' 000000  002067'
002074' 001000  000074     46880           INR     A               ;MAKE A NON-ZERO, SINCE NUMBER IS POSITIVE
                           46900                                   ; AND NON-ZERO, ROUND WILL EXIT WITH THE HO
                           46920                                   ; IN A, SO THE MSB WILL ALWAYS BE ZERO AND
                           46940                                   ; ADDING ONE WILL NEVER CAUSE A TO BE ZERO
002075' 001000  000315     46960           CALL    QINT            ;GET INTEGER PART IN C,D,E
002076' 000000  001372'
002077' 000000  002072'
002100' 001000  000315     46980           CALL    MOVFR           ;SAVE NUMBER IN FAC
002101' 000000  001225'
002102' 000000  002076'
                           47000   ;DECIDE IF THE NUMBER SHOULD BE PRINTED IN FIXED OR FLOATING NOTATION
002103' 001000  000001     47020           LXI     B,2*400+6+$CODE ;SET DECIMAL POINT COUNT FOR E NOTATION
002104' 000000  001006*
002105' 000000  002101'
                           47040                                   ;C = DIGIT COUNT
002106' 001000  000361     47060           POP     PSW             ;GET EXPONENT
                           47080           ADD     C               ;SEE IF NUMBER SHOULD BE PRINTED IN E NOTATION?
                                                                     SHOULD the
002107' 001000  000201
002110' 001000  000372     47100           JM      FOUT6           ;YES, IT IS .LT. .1
002111' 000000  002124'
002112' 000000  002104'
002113' 001000  000376     47120           CPI     7
002114' 000000  000007
002115' 001000  000322     47140           JNC     FOUT6           ;YES,  IT IS .GT. 999999
002116' 000000  002124'
002117' 000000  002111'
002120' 001000  000074     47160           INR     A
002121' 001000  000107     47180           MOV     B,A             ;B = DECIMAL POINT COUNT
002122' 001000  000076     47200           MVI     A,1             ;SET FIXED POINT FLAG, THE EXPONENT IS ZERO
002123' 000000  000001
                           47220                                   ; IF WE ARE USING FIXED POINT NOTATION
002124' 001000  000075     47240   FOUT6:  DCR     A               ;E NOTATION: ADD 5 TO ORIGINAL EXPONENT
002125' 001000  000341     47260           POP     H               ;GET BUFFER POINTER FROM STACK
002126' 001000  000365     47280           PUSH    PSW             ;SAVE EXPONENT FOR LATER
                           47300   ;CALCULATE THE DIGITS OF THE NUMBER
002127' 001000  000021     47320           LXI     D,FOUTBL        ;STORE LOC OF LARGEST POWER OF TEN
002130' 000000  002316'
002131' 000000  002116'
002132' 001000  000005     47340   FOUT8:  DCR     B               ;SEE IF IT IS TIME TO PRINT A DECIMAL POINT
002133' 001000  000066     47360           MVI     M,"."           ;PUT A DECIMAL POINT IN THE BUFFER
002134' 000000  000056
002135' 001000  000314     47380           CZ      INXHRT          ;INCREMENT THE BUFFER POINTER IF IT IS TIME
002136' 000000  001252'
002137' 000000  002130'
002140' 001000  000305     47400           PUSH    B               ;SAVE FLAGS
002141' 001000  000345     47420           PUSH    H               ;SAVE CHARACTER POINTER
```

```
2858  002142' 001000  000325          47440          PUSH    D               ;SAVE POWER OF TEN POINTER
2859  002143' 001000  000315          47460          CALL    MOVRF           ;GET NUMBER IN C,D,E
2860  002144' 000000  001240'
2861  002145' 000000  002136'
2862  002146' 001000  000341          47480          POP     H               ;GET POWER OF TEN POINTER
2863  002147' 001000  000006          47500          MVI     B,"0"-1         ;B = NEXT DIGIT TO BE PRINTED
2864  002150' 000000  000057
2865  002151' 001000  000004          47520  FOUT10: INR     B               ;ADD ONE TO DIGIT
2866  002152' 001000  000173          47540          MOV     A,E             ;SUBTRACT LO
2867  002153' 001000  000226          47560          SUB     M
2868  002154' 001000  000137          47580          MOV     E,A
2869  002155' 001000  000043          47600          INX     H               ;POINT TO NEXT BYTE OF POWER OF TEN
2870  002156' 001000  000172          47620          MOV     A,D             ;SUBTRACT MD
2871  002157' 001000  000236          47640          SBB     M
2872  002160' 001000  000127          47660          MOV     D,A
2873  002161' 001000  000043          47680          INX     H
2874  002162' 001000  000171          47700          MOV     A,C             ;SUBTRACT MO
2875  002163' 001000  000236          47720          SBB     M
2876  002164' 001000  000117          47740          MOV     C,A
2877  002165' 001000  000053          47760          DCX     H               ;POINT TO BEGINNING OF POWER OF TEN
2878  002166' 001000  000053          47780          DCX     H
2879  002167' 001000  000322          47800          JNC     FOUT10          ;SUBTRACT AGAIN IF RESULT WAS POSITIVE
2880  002170' 000000  002151'
2881  002171' 000000  002144'
2882  002172' 001000  000315          47820          CALL    FADDA           ;IT WASN'T, ADD POWER OF TEN BACK IN
2883  002173' 000000  000274'
2884  002174' 000000  002170'
2885  002175' 001000  000043          47840          INX     H               ;INCREMENT POINTER TO NEXT POWER OF TEN
2886  002176' 001000  000315          47860          CALL    MOVFR           ;SAVE C,D,E IN FAC
2887  002177' 000000  001225'
2888  002200' 000000  002173'
2889  002201' 001000  000353          47880          XCHG                    ;GET POWER OF TEN POINTER IN (DE)
2890  002202' 001000  000341          47900          POP     H               ;GET BUFFER POINTER
2891  002203' 001000  000160          47920          MOV     M,B             ;PUT CHARACTER IN BUFFER
2892  002204' 001000  000043          47940          INX     H               ;INCREMENT BUFFER POINTER
2893  002205' 001000  000301          47960          POP     B               ;GET COUNTERS OFF STACK
2894  002206' 001000  000015          47980          DCR     C               ;WAS THAT THE LAST DIGIT?
2895  002207' 001000  000302          48000          JNZ     FOUT8           ;DO MORE IF NOT
2896  002210' 000000  002132'
2897  002211' 000000  002177'
2898  002212' 001000  000005          48020          DCR     B               ;SEE IF DECIMAL POINT GOES AFTER LAST DIGIT
2899  002213' 001000  000312          48040          JZ      FOUT12          ;IT DOES, WE HAVE NO ZEROS TO SUPPRESS
2900  002214' 000000  002232'
2901  002215' 000000  002210'
2902                                   48060          ;SUPPRESS THE TRAILING ZEROS
2903  002216' 001000  000053          48080  FOUT11: DCX     H               ;GO BACK TO LAST CHARACTER
2904  002217' 001000  000176          48100          MOV     A,M             ;GET IT
2905  002220' 001000  000376          48120          CPI     "0"             ;IGNORE TRAILING ZEROS
2906  002221' 000000  000060
2907  002222' 001000  000312          48140          JZ      FOUT11
2908  002223' 000000  002216'
2909  002224' 000000  002214'
2910                                   48160          ;SUPPRESS DECIMAL POINT IF WE HAVE AN INTEGER
```



```
2911  002225' 001000  000376          48180          CPI     "."             ;IGNORE DECIMAL POINT BEFORE TRAILING ZEROS
2912  002226' 000000  000056
2913  002227' 001000  000304          48200          CNZ     INXHRT          ;IF NO DP, MOVE POINTER TO NEXT POSITION
2914  002230' 000000  001252'
2915  002231' 000000  002223'
2916  002232' 001000  000361          48220  FOUT12: POP     PSW             ;GET DECIMAL EXPONENT
2917  002233' 001000  000312          48240          JZ      FOUT17          ;RETURN IF NUMBER WAS IN FIXED POINT FORMAT
2918  002234' 000000  002271'
2919  002235' 000000  002230'
2920                                   48260          ;FLOATING POINT NOTATION -- PUT AN "E" IN THE BUFFER
2921  002236' 001000  000066          48280          MVI     M,"E"           ;PUT AN "E" IN THE BUFFER
2922  002237' 000000  000105
2923  002240' 001000  000043          48300          INX     H               ;PUT SIGN OF EXPONENT IN BUFFER
2924                                   48320          ;PUT IN THE SIGN OF THE EXPONENT
2925  002241' 001000  000066          48340          MVI     M,"+"           ;A PLUS IF POSITIVE
2926  002242' 000000  000053
2927  002243' 001000  000362          48360          JP      FOUT14
2928  002244' 000000  002252'
2929  002245' 000000  002234'
2930  002246' 001000  000066          48380          MVI     M,"-"           ;A MINUS IF NEGATIVE
2931  002247' 000000  000055
2932  002250' 001000  000057          48400          CMA                     ;NEGATE EXPONENT
2933  002251' 001000  000074          48420          INR     A
2934                                   48440          ;CALCULATE THE TWO DIGIT EXPONENT
2935  002252' 001000  000006          48460  FOUT14: MVI     B,"0"-1         ;INITIALIZE TEN'S DIGIT COUNT
2936  002253' 000000  000057
2937  002254' 001000  000004          48480  FOUT15: INR     B               ;INCREMENT DIGIT
2938  002255' 001000  000326          48500          SUI     12              ;SUBTRACT TEN
2939  002256' 000000  000012
2940  002257' 001000  000322          48520          JNC     FOUT15          ;DO IT AGAIN IF RESULT WAS POSITIVE
2941  002260' 000000  002254'
2942  002261' 000000  002244'
2943  002262' 001000  000306          48540          ADI     "0"+12          ;ADD BACK IN TEN AND CONVERT TO ASCII
2944  002263' 000000  000072
2945                                   48560          ;PUT THE EXPONENT IN THE BUFFER
2946  002264' 001000  000043          48580          INX     H
2947  002265' 001000  000160          48600          MOV     M,B             ;PUT TEN'S DIGIT OF EXPONENT IN BUFFER
2948  002266' 001000  000043          48620  FOUT19: INX     H               ;WHEN WE JUMP TO HERE, A IS ZERO
2949  002267' 001000  000167          48640          MOV     M,A             ;PUT ONE'S DIGIT IN BUFFER
2950  002270' 001000  000043          48660          INX     H               ;INCREMENT POINTER
2951  002271' 001000  000161          48680  FOUT17: MOV     M,C             ;PUT ZERO AT END OF BUFFER
2952  002272' 001000  000341          48700          POP     H               ;EXIT WITH (HL) POINTING TO STRING
2953  002273' 001000  000311          48720          RET                     ;ALL DONE
2954
2955                                   48760          ;SEE IF FAC ,LE, 999999,499
2956  002274' 001000  000001          48780  FOUTC8: MOVRI   224,164,043,367 ;COMPARE NUMBER WITH CONSTANT
2957  002275' 000000  000164
2958  002276' 000000  000224
2959  002277' 001000  000021
2960  002300' 000000  000367
2961  002301' 000000  000043
2962  002302' 001000  000315          48800          CALL    FCOMP
2963  002303' 000000  001317'
```

```
 2964  002304' 000000  002260'
 2965  002305' 001000  000341          48820          POP     H               ;GET RETURN ADDRESS OFF STACK
 2966  002306' 001000  000342          48840          JPO     FOUT9           ;NUMBER TOO BIG, DIVIDE BY TEN
 2967  002307' 000000  002260'
 2968  002310' 000000  002303'
 2969  002311' 001000  000351          48860          PCHL                    ;NUMBER OK, RETURN
 2970
 2971                                   48900          ;CONSTANTS FOR FOUT
 2972  002312' 000000  000000          48920  FHALF:  000     ; 1/2
 2973  002313' 000000  000000          48940          000     ;THIS CONSTANT IS ALSO USED BY SQR, SIN, COS
 2974  002314' 000000  000000          48960          000
 2975  002315' 000000  000200          48980          200
 2976                                   49000          ;POWER OF TEN TABLE
 2977  002316' 000000  000240          49020  FOUT8L: 240     ; 100000
 2978  002317' 000000  000206          49040          206
 2979  002320' 000000  000001          49060          001
 2980  002321' 000000  000020          49080          020     ; 10000
 2981  002322' 000000  000047          49100          047
 2982  002323' 000000  000000          49120          000
 2983  002324' 000000  000350          49140          350     ; 1000
 2984  002325' 000000  000003          49160          003
 2985  002326' 000000  000000          49180          000
 2986  002327' 000000  000144          49200          144     ; 100
 2987  002330' 000000  000000          49220          000
 2988  002331' 000000  000000          49240          000
 2989  002332' 000000  000012          49260          012     ; 10
 2990  002333' 000000  000000          49280          000
 2991  002334' 000000  000000          49300          000
 2992  002335' 000000  000001          49320          001     ; 1
 2993  002336' 000000  000000          49340          000
 2994  002337' 000000  000000          49360          000>
 2995                                   49380  IFE     LENGTH=2,<
 2996                                   49400          ;OUTPUT THE VALUE IN THE FAC ACCORDING TO THE FORMAT SPECIFICATIONS
 2997                                   49420          ; IN A,B,C
 2998                                   49440          ;ALL REGISTERS ARE ALTERED
 2999                                   49460          ;THE ORIGINAL CONTENTS OF THE FAC IS LOST
 3000                                   49480
 3001                                   49500          ;THE FORMAT IS SPECIFIED IN A, B AND C AS FOLLOWS:
 3002                                   49520          ;THE BITS OF A MEAN THE FOLLOWING:
 3003                                   49540  ;BIT 7  0 MEANS FREE FORMAT OUTPUT, I.E, THE OTHER BITS OF A MUST BE ZERO,
 3004                                   49560  ;       TRAILING ZEROS ARE SUPPRESSED, A NUMBER IS PRINTED IN FIXED OR FLOATING
 3005
 3006                                   49580  ;       POINT NOTATION ACCORDING TO ITS MAGNITUDE, THE NUMBER IS LEFT
 3007                                   49600  ;       JUSTIFIED IN ITS FIELD, B AND C ARE IGNORED,
 3008                                   49620  ;       1 MEANS FIXED FORMAT OUTPUT, I.E, THE OTHER BITS OF A ARE CHECKED FOR
 3009                                   49640  ;       FORMATTING INFORMATION, THE NUMBER IS RIGHT JUSTIFIED IN ITS FIELD,
 3010                                   49660  ;       TRAILING ZEROS ARE NOT SUPPRESSED,  THIS IS USED FOR PRINT USING,
 3011                                   49680  ;BIT 6  1 MEANS GROUP THE DIGITS IN THE INTEGER PART OF THE NUMBER INTO GROUPS
 3012                                   49700  ;       OF THREE AND SEPARATE THE GROUPS BY COMMAS
 3013                                   49720  ;       0 MEANS DON'T PRINT THE NUMBER WITH COMMAS
 3014                                   49740  ;BIT 5  1 MEANS FILL THE LEADING SPACES IN THE FIELD WITH ASTERISKS ("*")
 3015                                   49760  ;BIT 4  1 MEANS OUTPUT THE NUMBER WITH A FLOATING DOLLAR SIGN ("$")
 3016                                   49780  ;BIT 3  1 MEANS PRINT THE SIGN OF A POSITIVE NUMBER AS A PLUS SIGN ("+")
```



```
 3017                                   49800  ;       INSTEAD OF A SPACE
 3018                                   49820  ;BIT 2  1 MEANS PRINT THE SIGN OF THE NUMBER AFTER THE NUMBER
 3019                                   49840  ;BIT 1  UNUSED
 3020                                   49860  ;BIT 0  1 MEANS PRINT THE NUMBER IN FLOATING POINT NOTATION I,E, "E NOTATION"
 3021                                   49880  ;       IF THIS BIT IS ON, THE COMMA SPECIFICATION (BIT 6) IS IGNORED,
 3022                                   49900  ;       0 MEANS PRINT THE NUMBER IN FIXED POINT NOTATION,  NUMBER ,GE, 1E16
 3023                                   49920  ;       CANNOT BE PRINTED IN FIXED POINT NOTATION,
 3024                                   49940
 3025                                   49960  ;B AND C TELL HOW BIG THE FIELD IS:
 3026                                   49980  ;B  =   THE NUMBER OF PLACES IN THE FIELD TO THE LEFT OF THE DECIMAL POINT
 3027                                   50000  ;       (B DOES NOT INCLUDE THE DECIMAL POINT)
 3028                                   50020  ;C  =   THE NUMBER OF PLACES IN THE FIELD TO THE RIGHT OF THE DECIMAL POINT
 3029                                   50040  ;       (C INCLUDES THE DECIMAL POINT)
 3030                                   50060  ;       B AND C DONOT INCLUDE THE 4 POSITIONS FOR THE EXPONENT IF BIT 0 IS ON
 3031                                   50080          ;FOUT ASSUMES B+C ,LE, 24 (DECIMAL)
 3032                                   50100
 3033                                   50120          ;ENTRY TO PRINT THE FAC IN FREE FORMAT
 3034  002340' 257     000000  FOUT:   50140          XRA     A               ;SET FORMAT FLAGS TO FREE FORMATED OUTPUT
 3035                                   50160          ;ENTRY TO PRINT THE FAC USING THE FORMAT SPECIFICATIONS IN A, B AND C
 3036  002341' 315     000000  PUFOUT: 50180          CALL    FOUINI          ;SAVE THE FORMAT SPECIFICATION IN A AND PUT
 3037                                   50200                                  ;A SPACE FOR POSITIVE NUMBERS IN THE BUFFER
 3038  002344' 305     000000          50220          PUSH    B               ;SAVE THE FIELD LENGTH SPECIFICATIONS
 3039  002345' 346     000             50240          ANI     10              ;CHECK IF POSITIVE NUMBERS GET A PLUS SIGN
 3040  002347' 312     000000          50260          JZ      FOUT1           ;THEY DON'T
 3041  002352' 076     053             50280          MVI     M,"+"           ;THEY DO, PUT IN A PLUS SIGN
 3042  002354' 072     000000  FOUT1:  50300          LDA     VALTYP          ;SEE WHAT KIND OF A VALUE WE HAVE
 3043  002357' 107                     50320          MOV     B,A             ;SAVE IT
 3044  002360' 353                     50340          XCHG                    ;SAVE BUFFER POINTER
 3045  002361' 315     000000          50360          CALL    VSIGN           ;GET THE SIGN OF THE FAC
 3046  002364' 353                     50380          XCHG                    ;PUT THE BUFFER POINTER BACK IN (HL)
 3047  002365' 170                     50400          MOV     A,B             ;GET THE VALTYP BACK
 3048  002366' 362     000000          50420          JP      FOUT2           ;IF WE HAVE A NEGATIVE NUMBER, NEGATE IT
 3049  002371' 076     055             50440          MVI     M,"-"           ; AND PUT A MINUS SIGN IN THE BUFFER
 3050  002373' 345                     50460          PUSH    H               ;SAVE THE BUFFER POINTER
 3051  002374' 315     000000          50480          CALL    VNEG            ;NEGATE THE NUMBER
 3052  002377' 341                     50500          POP     H               ;GET THE BUFFER POINTER BACK
 3053  002400' 043             FOUT2:  50520          INX     H               ;POINT TO WHERE THE NEXT CHARACTER GOES
 3054  002401' 072     000000          50540          LDA     TEMP3           ;GET THE FORMAT SPECIFICATION
 3055  002404' 127                     50560          MOV     D,A             ;SAVE IT FOR LATER
 3056  002405' 027                     50580          RAL                     ;PUT THE FREE FORMAT OR NOT BIT IN THE CARRY
 3057  002406' 072     000000          50600          LDA     VALTYP          ;GET THE VALTYP, VNEG COULD HAVE CHANGED THIS
 3058                                   50620                                  ; SINCE -32768 IS INT AND 32768 IS SNG,
 3059                                   50640                                  ; SO B IS NOT ACCURATE
 3060  002411' 332     000000          50660          JC      FOUTFX          ;THE MAN WANTS FIXED FORMATED OUTPUT
 3061                                   50680          ;HERE TO PRINT NUMBERS IN FREE FORMAT
 3062  002414' 341                     50700          POP     B               ;WE CAN IGNORE THE OLD B AND C
 3063  002415' 076     060             50720          MVI     M,"0"           ;PUT A ZERO IN THE BUFFER IN CASE THE NUMBER
 3064  002417' 312     000000          50740          JZ      FOUTZR          ; IS ZERO,  IT IS, FINISH IT UP
 3065  002422' 376     004             50760          CPI     4               ;DECIDE WHAT KIND OF A VALUE WE HAVE
 3066  002424' 322     000000          50780          JNC     FOUFRY          ;WE HAVE A SNG OR DBL
 3067                                   50800          ;HERE TO PRINT AN INTEGER IN FREE FORMAT
 3068  002427' 041     000000          50820          LXI     B,$CODE         ;SET THE DECIMAL POINT COUNT AND COMMA COUNT
 3069                                   50840                                  ; TO ZERO
```