

F4 MAC 23-AUG-64 06108 FLOWING POINT INPUT ROUTINE

```

2365 001574* 001000 000345      39580  PUSH H           ;SAVE TEXT POINTER
2366 001575* 001000 002041      39680  LXI M,FINEC    ;PUT FINEC ON STACK SO WE CAN JUMP
2367 001577* 000000 001621*      39680
2368 001577* 000000 001571*      39680
2369 001600* 001000 000343      39680  XTHL>          ;TO IT IN LESS BYTES
2370 001601* 001000 000025      39740  DCR D           ;SET SIGN OF EXPONENT FLAG
2371 001602* 001000 000376      39760  CPI MINUTK    ;NEGATIVE EXPONENT?
2372 001603* 000000 000000*      39680
2373
2374
2375
2376 001604* 001000 000310      39780  IFE STRING,<
2377 001605* 001000 000376      39780  JZ FINEC>
2378 001606* 001000 000055      39780  RZ
2379 001607* 001000 000310      39788  CPI *=H
2380
2381
2382 001611* 001000 000376      39820  IFN STRING,<
2383 001612* 000000 000053      39840  CPI *=H
2384 001613* 001000 000310      39860  RZ>
2385 001614* 001000 000376      39880  CPI PLUSTK    ;IGNORE "+"
2386 001615* 000000 000000*      39900  IFE STRING,<
2387
2388
2389 001616* 001000 000310      39940  JZ FINEC>
2390 001617* 001000 000361      39960  RZ
2391 001620* 001000 000053      40000  POP PSW>      ;GET FINEC OFF STACK
2392
2393
2394 001621* 000000 000327      40040  DCX H           ;CHECK IF LAST CHARACTER WAS A DIGIT
2395
2396 001622* 001000 000032      40080  HERE TO GET THE NEXT DIGIT OF THE EXPONENT
2397 001623* 000000 001744*      40080  FINEC: CHRET    ;GET NEXT CHARACTER
2398 001624* 002000 001576*      40080  JC FINEC0      ;IS IT A DIGIT?
2399 001625* 001000 000024      40080  INR D           ;IN, EXPONENT ALL IN
2400 001626* 001000 000382      40100  JNZ FINE       ;SET ITS SIGN
2401 001627* 002000 001641*      40100
2402 001631* 001000 000257      40120  XRA A           ;MAKE SURE C IS NOT 377
2403 001632* 001000 000225      40120  SUB E           ;SUB E
2404 001633* 001000 000137      40140  MOV E,A         ;MOV E,A
2405 001634* 001000 000014      40180  INR C           ;IN, EXPONENT
2406
2407
2408 001635* 001000 000014      40200  HERE TO CHECK IF WE HAVE SEEN 2 DECIMAL POINTS AND SET THE DECIMAL
2409 001636* 001000 000312      40240  I POINT FLAG
2410 001637* 000000 001555*      40260  FINDP: INR C   ;DECIMAL POINTS!! == SET FLAG
2411 001640* 000000 001627*      40260  JZ FINEC        ;CONTINUE SCANNING CHARACTERS
2412
2413
2414
2415 001641* 001000 000345      40300  HERE TO MULTIPLY OR DIVIDE BY 10 THE CURRENT NUMBER OF TIMES,
2416 001642* 001000 000173      40320  JNE HAVE ALREADY READ IN ALL THE DIGITS.
2417 001643* 001000 000220      40340  FINE1: PUSH H   ;SAVE POINTER FOR LATER
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470

```

F4 MAC 23-AUG-64 06108 FLOWING POINT INPUT ROUTINE

```

2416 001644* 001000 000364      40400  FINE2: CP FINMUL  ;MULTIPLY BY TEN IF EXPONENT IS POSITIVE
2417 001645* 000000 001672*      40400
2418 001646* 000000 001637*      40420  JP FINE3      ;DIVIDE BY TEN IF EXPONENT IS NEGATIVE
2419 001647* 001000 000362      40420
2420 001648* 000000 001666*      40440
2421 001649* 001000 000362      40440
2422 001650* 000000 001655*      40440
2423 001651* 000000 001655*      40440
2424 001652* 001000 000315      40440  PUSH PSH        ;SAVE EXPONENT
2425 001653* 001000 000315      40460  CALL DIV10    ;DIVIDE NUMBER BY TEN
2426 001654* 000000 000637*      40460
2427 001655* 000000 001650*      40480
2428 001656* 001000 000361      40480  PDP PSW        ;GET EXPONENT
2429 001657* 001000 000074      40500  INR A           ;INCREMENT IT
2430 001658* 001000 000302      40520  FINE3: JNZ FINE2  ;DO AGAIN IF WE ARE NOT DONE
2431 001659* 000000 001644*      40520
2432 001660* 000000 001654*      40540  IFE STRING,<
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470

```

```

    2471          41260   RZ
    2472          41260   INR D           ;EXONENT IS STILL POSITIVE, RESET FLAG
    2473          41300   CPI PLUSTK      ;IGNORE A LEADING PLUS SIGN
    2474          41320   RZ
    2475          41360   CPI #+
    2476          41360   RZ
    2477          41360   DCX H           ;THE FIRST CHARACTER WAS NOT A SIGN, GO BACK
    2478          41400   POP PSW          ;AND CHECK FOR A DIGIT
    2479          41420   JNE FINED        ;POP FINC OFF THE STACK, WE NO LONGER NEED IT
    2480          41440   THERE TO GET THE NEXT DIGIT OF THE EXPONENT
    2481          41460   FINED: CHRGET      ;GET THE NEXT CHARACTER
    2482          41460   JNE FINED        ;PACK THE NEXT DIGIT INTO THE EXPONENT
    2483          41480   INR D           ;IT MAY NOT BE A DIGIT, PUT THE CORRECT SIGN ON
    2484          41480   JNZ FINE         ;THE EXPONENT, IT IS POSITIVE
    2485          41480   XRA A           ;THE EXPONENT IS NEGATIVE
    2486          41480   SUB E           ;NEGATE IT
    2487          41480   MOV E,A          ;SAVE IT AGAIN
    2488          41480   THERE TO FINISH UP THE NUMBER
    2489          41480   FINE: LDA VALTYP      ;FINISH UP -- WHAT KIND OF A NUMBER IS IT?
    2490          41480   CPI 2
    2491          41480   JNE FINEP        ;IT IS A FLOATING POINT ONE
    2492          41480   THERE TO FINISH UP AN INTEGER
    2493          41480   FINEP: POP PSW          ;IT IS AN INTEGER, GET ITS SIGN
    2494          41480   JNE FINE          ;IT IS AN INTEGER, GET ITS SIGN
    2495          41480   XCHG PSW          ;SAVE THE TEXT POINTER IN (DE)
    2496          41480   CZ INEG          ;NEGATE IT IF NECESSARY
    2497          41480   XCHG PSW          ;GET THE TEXT POINTER BACK IN (HL)
    2498          41480   RET             ;ALL DONE
    2499          41480   THERE TO FINISH UP A FLOATING POINT NUMBER
    2500          41480   FINEF: PUSH H           ;SAVE THE TEXT POINTER
    2501          41480   MUL A,E          ;MULTIPLY BY TEN TIMES WE HAVE TO MULTIPLY
    2502          41480   SUB B             ;BY DIVIDE BY TEN
    2503          41480   THERE TO MULTIPLY OR DIVIDE BY TEN, THE CORRECT NUMBER OF TIMES
    2504          41480   FINEF2: CP FINMUL      ;MULTIPLY IF WE HAVE TO
    2505          41480   JNE FINEF2       ;DIVIDE IF WE HAVE TO
    2506          41480   CM FINDIV      ;MULTIPLY OR DIVIDE AGAIN IF WE ARE NOT DONE
    2507          41480   JNE FINEF2
    2508          41480   THERE TO PUT THE CORRECT SIGN ON THE NUMBER
    2509          41480   POP D             ;GET THE TEXT POINTER
    2510          41480   PUP PSW          ;GET THE SIGN
    2511          41480   CZ NEG           ;NEGATE IT IF NECESSARY
    2512          41480   XCHG PSW          ;GET THE TEXT POINTER IN (HL)
    2513          41480   LUA VALTYP      ;WE WANT -32768 TO BE AN INT, BUT UNTIL NOW
    2514          41480   CPI 4             ;IT WOULD BE A SNG
    2515          41480   RNZ              ;IT IS NOT SNG, SO IT IS NOT -32768
    2516          41480   PUSH H           ;WE HAVE A SNG, SAVE TEXT POINTER
    2517          41480   LXI H,POPHRT      ;GET ADDRESS THAT POP'S H OFF STACK BECAUSE
    2518          41480   PUSH H           ;CODE DOES ANYTHING WITH THE STACK
    2519          41480   CALL CUNIS2      ;CHECK IF WE HAVE -32768
    2520          41480   RET              ;WE DON'T, POPHRT IS STILL ON THE STACK SO
    2521          41480   ;WE CAN JUST RETURN
    2522          42200   THERE TO CHECK IF WE HAVE SEEN 2 DECIMAL POINTS AND SET THE DECIMAL
    2523          42200   POINT FLAG
    42300   FINDPD: INR C           ;SET THE FLAG

```

```

    2524          42320   JNZ FINEC      ;WE HAD 2 DECIMAL POINTS, NOW WE ARE DONE
    2525          42340   CALL FINFRC      ;THIS IS THE FIRST ONE, CONVERT FAC TO SNG
    2526          42360   JMP FINC          ;CONTINUE LOOKING FOR DIGITS
    2527          42380
    2528          42400   ;FORCE THE FAC TO BE SNG OR DBL
    2529          42420   ;IF THE ZERO FLAG IS ON, THEN FORCE THE FAC TO BE SNG
    2530          42440   ;IF THE ZERO FLAG IS OFF, FORCE THE FAC TO BE DBL
    2531          42460   FINFRC: PUSH H           ;SAVE TEXT POINTER
    2532          42480   PUSH D             ;SAVE EXPONENT INFORMATION
    2533          42500   PUSH B             ;SAVE DECIMAL POINT INFORMATION
    2534          42520   PUSH PSW          ;SAVE WHAT WE WANT THE FAC TO BE
    2535          42540   CPI FCKSNG      ;CONVERT TO SNG IF WE HAVE TO
    2536          42560   POP PSW          ;DETERMINE FLAG BACK
    2537          42580   CNZ FRCDBL      ;CONVERT TO DBL IF WE HAVE TO
    2538          42600   POP B             ;GET DECIMAL POINT INFORMATION BACK
    2539          42620   POP D             ;GET EXPONENT INFORMATION BACK
    2540          42640   POP H             ;GET TEXT POINTER BACK
    2541          42660   RET              ;ALL DONE
    2542          42700   ;THIS SUBROUTINE MULTIPLIES BY TEN ONCE,
    2543          42720   ;IT IS A SUBROUTINE BECAUSE IT SAVES BYTES WHEN WE CHECK IF A IS ZERO
    2544          001672* 001000 000310 42740   FINMUL: RZ             ;RETURN IF EXPONENT IS ZERO, ENTRY FROM FOUT
    2545          001673* 001000 000365 42760   FINMLT: PUSH PSW          ;SAVE EXPONENT, ENTRY FROM FOUT
    2546          42780   IFN LENGTH=2,<
    2547          42800   CALL MUL10>    ;MULTIPLY BY TEN
    2548          001674* 001000 000315 42820   IFE LENGTH=2,<
    2549          001675* 000000 001106* 42840   LUA VALTYP      ;SEE WHAT KIND OF NUMBER WE HAVE
    2550          001676* 000000 001666* 42860   CPI 4             ;SAVE THE TYPE
    2551          42880   PUSH PSW          ;WE HAVE A SNG, MULTIPLY BY 10.0
    2552          42900   CZ MUL10      ;WE HAVE A SNG, MULTIPLY BY 10.0
    2553          42920   POP PSW          ;GET THE TYPE BACK
    2554          42940   CNZ DMUL10>    ;WE HAVE A DBL, MULTIPLY BY 1000
    2555          42960   POP PSW          ;GET EXPONENT
    2556          001677* 001000 000361 42980   DCRART: DCX A          ;DECREASE IT
    2557          001700* 001000 000267 43000   RET              ;ALL DONE
    2558          001701* 001000 000311 43000   IFE LENGTH=2,<
    2559          43020   PUSH PSW          ;WE HAVE TO DIVIDE -- SAVE COUNT
    2560          43040   LUA VALTYP      ;SEE WHAT KIND OF NUMBER WE HAVE
    2561          43060   CPI 4             ;SAVE THE TYPE
    2562          43080   PUSH PSW          ;WE HAVE A SNG NUMBER
    2563          43100   CPI 4             ;WE HAVE A SNG NUMBER
    2564          43120   PUSH PSW          ;GET THE TYPE BACK
    2565          43140   CPI DIV10      ;WE HAVE A DBL NUMBER
    2566          43160   POP PSW          ;GET COUNT BACK
    2567          43180   CNZ DIV10>    ;UPDATE IT
    2568          43200   POP PSW          ;GET COUNT BACK
    2569          43220   INR A           ;UPDATE IT
    2570          43240   RET>
    2571          43260   ;HERE TO PACK THE NEXT DIGIT OF THE NUMBER INTO THE FAC
    2572          43280   ;WE MULTIPLY THE FAC BY TEN AND ADD IN THE NEXT DIGIT
    2573          43300   FINDIG: INR C           ;SET THE FLAG

```

THIS subroutine multiplies by
TEN ONCE
IT is used by FINP, FOUT

```

2577          43340  IFN   LENGTH=2,<
2578  001702f  001000  000325  43340  PUSH   D           !DIGITS: SAVE EXPONENT INFORMATION
2579  001703f  001000  000127  43380  MOV    D,A         !PROTECT DIGIT FROM BELOW
2580  001704f  001000  000170  43400  MOV    A,B         !INCREMENT DECIMAL PLACE COUNT
2581  001705f  001000  000211  43420  ADC    C           ; IF PAST THE DECIMAL POINT
2582  001706f  001000  000167  43440  MOV    B,A         !SAVE NECESSARY DATA
2583  001707f  001000  000305  43460  PUSH   H           !SAVE DIGIT
2584  001708f  001000  000135  43480  PUSH   M           !SAVE DIGIT
2585  001711f  001000  000325  43500  PUSH   D           !SAVE DIGIT
2586  001712f  001000  000315  43520  CALL   MUL10        !MULTIPLY OLD NUMBER BY 10
2587  001713f  002000  001106f  43540  POP    PSW          !GET NEXT DIGIT
2588  001714f  002000  001675f  43540  SUI   "#0"        !SUBTRACT OFF ASCII CODE
2589  001715f  001000  000361  43560  IFE   EXTFNC,<      !INPUT NUMBER ON STACK
2590  001716f  001000  000326  43560  CALL   PUSHF        !CONVERT TO FLOATING POINT NUMBER
2591  001717f  000000  000060  43580  POP    D           !ADD IN NEXT DIGIT
2592          43580  IFE   EXTFNC,<      !RECALL DATA
2593  001721f  000000  001713f  43590  CALL   ADDD>>       !ADD IN NEXT CHARACTER
2594          43590  IFE   EXTFNC,<      !GET NEXT CHARACTER
2595  001723f  001000  000341  43600  CALL   ADDD>>       !GET NEXT CHARACTER
2596          43600  IFE   EXTFNC,<      !GET NEXT CHARACTER
2597  001725f  001000  000321  43620  CALL   FINLOG        !FINISH LOGIC
2598  001726f  001000  000363  43640  POP    H           !RECALL DATA
2599  001727f  002000  001555f  43660  POP    B           !RECALL DATA
2600  001730f  000000  001721f  43680  POP    D           !RECALL DATA
2601          43680  IFE   LENGTH=2,<
2602  001725f  001000  000341  43700  PUSH   H           !SAVE EXPONENT INFORMATION
2603  001726f  001000  000321  43720  POP    B           !INCREMENT DECIMAL PLACE COUNT IF WE ARE
2604  001727f  001000  000363  43740  POP    D           !PAST THE DECIMAL POINT
2605          43740  IFE   LENGTH=2,<
2606  001726f  001000  000363  43760  PUSH   B           !SAVE DECIMAL POINT INFORMATION
2607  001727f  001000  000363  43780  POP    D           !SAVE TEXT POINTER
2608  001728f  001000  000340  43800  PUSH   A,M         !GET THE DIGIT
2609  001729f  001000  000340  43820  SUI   "#0"        !CONVERT IT TO ASCII
2610  00172af  001000  000340  43840  PUSH   PSH          !SAVE THE DIGIT
2611          43840  IFE   LENGTH=2,<
2612  00172bf  001000  000340  43860  ADC    C           !SEE WHAT KIND OF A NUMBER WE HAVE
2613  00172cf  001000  000340  43880  MOV    B,A         !CHECK FOR OVERFLOW
2614  00172df  001000  000340  43900  PUSH   B           !JUMP IF OVERFLOW OCCURRED
2615  00172ef  001000  000340  43920  POP    H           !WE DO NOT HAVE AN INTEGER
2616  00172ff  001000  000340  43940  PUSH   A,M         !HERE TO PACK THE NEXT DIGIT OF AN INTEGER
2617  00172af  001000  000340  43960  LHL0   FACTL        !WE HAVE AN INTEGER, GET IT IN (HL)
2618  00172bf  001000  000340  43980  LXI   D,"03277->CODE" !SEE IF WE WILL OVERFLOW
2619  00172cf  001000  000340  44000  COMPAR        !COMPAR RETURNS WITH CARRY ON IF
2620  00172df  001000  000340  44020  JNC   FIND62        !THE NUMBER IS TOO BIG
2621  00172ef  001000  000340  44040  JNC   FIND62        !IF (HL).LT. (DE), SO THE NUMBER IS TOO BIG
2622  00172ff  001000  000340  44060  MOV   D,M         !COPY (HL) INTO (DE)
2623  00172af  001000  000340  44080  MOV   E,L         !MULTIPLY (HL) BY 2
2624  00172bf  001000  000340  44100  DAD   H           !MULTIPLY (HL) BY 2, (HL) NOW IS 4*(DE)
2625  00172cf  001000  000340  44120  DAD   M           !ADD IN OLD (HL) TO GET 5*(DE)
2626  00172df  001000  000340  44140  DAD   D           !
2627  00172ef  001000  000340  44160  !
2628  00172ff  001000  000340  44180  !
2629  00172af  001000  000340  44200  !

```

```

2630          44280  DAD   H           !MULTIPLY BY 2 TO GET TEN TIMES THE OLD (HL)
2631  001728f  001000  000340  44280  POP    PSW          !GET THE DIGIT
2632  001729f  001000  000340  44300  MOV   C,A         !SAVE IT SO WE CAN USE DAD, B IS ALREADY ZERO
2633  00172af  001000  000340  44320  DAD   B           !ADD IN THE NEXT DIGIT
2634  00172bf  001000  000340  44340  MOV   A,M         !CHECK FOR OVERFLOW
2635  00172cf  001000  000340  44360  ORA   C           !JUMP IF OVERFLOW OCCURRED
2636  00172df  001000  000340  44380  JR    FIND61        !EVERYTHING IS FINE, STORE THE NEW NUMBER
2637  00172ef  001000  000340  44400  SHLD   FACTL        !ALL DONE, GET TEXT POINTER BACK
2638  00172ff  001000  000340  44420  FIND61: POP   H           !
2639  00172af  001000  000340  44440  POP   B           !GET DECIMAL POINT INFORMATION BACK
2640  00172bf  001000  000340  44460  POP   D           !GET EXPONENT INFORMATION BACK
2641  00172cf  001000  000340  44480  JMP   FINC          !GET THE NEXT CHARACTER
2642  00172df  001000  000340  44500  !
2643  00172ef  001000  000340  44520  !
2644  00172ff  001000  000340  44540  !
2645  00172af  001000  000340  44560  !
2646  00172bf  001000  000340  44580  !
2647  00172cf  001000  000340  44600  !
2648  00172df  001000  000340  44620  !
2649  00172ef  001000  000340  44640  !
2650  00172ff  001000  000340  44660  !
2651  00172af  001000  000340  44680  !
2652  00172bf  001000  000340  44700  !
2653  00172cf  001000  000340  44720  !
2654  00172df  001000  000340  44740  !
2655  00172ef  001000  000340  44760  !
2656  00172ff  001000  000340  44780  !
2657  00172af  001000  000340  44800  !
2658  00172bf  001000  000340  44820  !
2659  00172cf  001000  000340  44840  !
2660  00172df  001000  000340  44860  !
2661  00172ef  001000  000340  44880  !
2662  00172ff  001000  000340  44900  !
2663  00172af  001000  000340  44920  !
2664  00172bf  001000  000340  44940  !
2665  00172cf  001000  000340  44960  !
2666  00172df  001000  000340  45000  !
2667  00172ef  001000  000340  45020  !
2668  00172ff  001000  000340  45040  !
2669  001731f  001000  000315  45080  IFN   EXTFNC,<      !SUBROUTINE FOR FIN, LOG
2670  001732f  000000  001205f  45080  FINLOG: CALL  PUSHF        !SAVE FAC ON STACK
2671  001733f  000000  001727f  45080  !
2672  001734f  001000  000315  45080  CALL   FLOAT        !CONVERT A TO A FLOATING POINT NUMBER
2673  001735f  000000  001150f  45080  LENGTH=2,<
2674  001736f  000000  001732f  45080  JNP   FADD>>       !ADD IT IN
2675          45080  IFN   EXTFNC,<
2676  001737f  001000  000303  45100  LENGTH=2,<
2677  001738f  002000  000023f  45100  JNP   FADD>>       !GET PREVIOUS NUMBER OFF STACK
2678  001739f  000000  001735f  45100  JNP   FADD>>       !ADD IT IN
2679          45120  IFE   LENGTH=2,<
2680  001738f  002000  000023f  45140  !
2681  001739f  000000  001735f  45160  !

```

F4 MAC 23-AUG-64 06108

```

2683          45200      HERE WE PACK IN THE NEXT DIGIT OF THE EXPONENT
2684          45220      ;THE MULIPLY THE OLD EXPONENT BY TEN AND ADD IN THE NEXT DIGIT
2685          45240      ;NOTE: EXPONENT OVERFLOW IS NOT CHECKED FOR
2686 001742' 001000 000175 45260  FINEDD: MOV A,E      ;EXPONENT DIGIT == MULTIPLY EXPONENT BY 10
2687 001743' 001000 000087 45280  RLC A           ;FIRST BY 4
2688 001744' 001000 000087 45300  ADD E           ;ADD 1 TO MAKE 5
2689 001745' 001000 000203 45320  RLC             ;NOW DOUBLE TO GET 10
2690 001746' 001000 000087 45340  ADD M           ;ADU IT IN
2691 001747' 001000 000206 45360  SUI "#0"        ;SUBTRACT OFF ASCII CODE
2692 001758' 001000 000326 45380
2693 001751' 000000 000000
2694 001752' 001000 000137 45400  MUV E,A       ;STORE EXPONENT
2695 001753' 001000 000503 45420  JMP FINEC     ;CONTINUE
2696 001754' 000000 001621'
2697 001755' 000000 001742'
2698
2699          45440  PAGE

```

F4 MAC 23-AUG-64 06108

FLOATING POINT OUTPUT ROUTINE

```

2700          45460  SUBTTL FLOATING POINT OUTPUT ROUTINE
2701          45480  ;ENTRY TO LINPRT
2702 001756' 001000 000345 45500  INPRHT: PUSH H      ;SAVE LINE NUMBER
2703          45520  LXI H,INTXT#  ;PRINT MESSAGE
2704 001757' 001000 000041
2705 001758' 001000 000000
2706 001759' 001000 001754'
2707 001760' 001000 000315 45540  CALL STROUT
2708 001761' 000000 000000*
2709 001762' 000000 001760'
2710 001763' 001000 000341 45560  PDP H           ;FALL INTO LINPRT
2711
2712 001764' 000000 000000
2713 001765' 000000 000000
2714
2715 001766' 001000 000353 45580  LINPRT: IFN LENGTH=2,<
2716 001767' 001000 000257 45600  XCHG             ;SET UP REGISTERS FOR FLOATR
2717 001768' 000000 000006 45620  XRA A
2718 001769' 000000 000230 45640  MVI B,230
2719 001770' 001000 000315 45660  CALL FLOATR    ;CONVERT TO FLOATING POINT
2720 001771' 000000 000155'
2721 001772' 000000 001155' 45680  IFE LENGTH=2,<
2722 001773' 000000 001763' 45700  IFE LENGTH=2,<
2723 001774' 000000 000000
2724 001775' 000000 000000
2725 001776' 000000 000000
2726 001777' 000000 000000*
2727 001778' 000000 001773'
2728 002000' 001000 000345 45780  PUSH H>        ;RETURN TO IT AND DO AN "INX H"
2729          45800  CALL CUNISS   ;PUT THE LINE NUMBER IN THE FAC AS AN INTEGER
2730          45820  XRA A         ;SET FORMAT TO FREE FORMAT
2731          45840  CALL FOUINI#  ;SET UP THE SIGN
2732          45860  LXI H,SIRUOI#  ;PUT PRINT STRING ADDRESS ON STACK SO WE WILL
2733          45880  PUSH H>        ;THE BEGINNING OF A LINE NUMBER
2734          45900  ;THIS GETS RID OF THE SPACE FOR THE SIGN AT
2735          45920  ;THE BEGINNING OF THE NUMBER
2736          45940  ;FALL INTO FOUT
2737          45960  IFE LENGTH=2,<
2738          45980  PUSH H>        ;PUT HUMMY FIELD LENGTHS ON STACK
2739          46000  JMP FOUT1#    ;PRINT THE NUMBER
2740
2741          46020  ;FLOATING OUTPUT OF FAC
2742          46040  ;ALTERS ALL REGISTERS
2743          46060  ;THE ORIGINAL CONTENTS OF THE FAC IS LOST
2744 002001' 001000 000041 46120  IFN LENGTH=2,<
2745 002002' 000000 000000*
2746 002003' 000000 001775'
2747 002004' 001000 000345 46140  FOUT1: LXI H,FBUFFR  ;GET BEGINNING OF CHARACTER BUFFER
2748          46160  PUSH H           ;SAVE IT FOR WHEN WE RETURN
2749          46180  ;PUT THE SIGN OF THE NUMBER IN THE BUFFER AND MAKE IT POSITIVE
2750          46200  FSIGN            ;GET SIGN OF NUMBER
2751 002005' 001000 000000 46220  MVI M," "        ;PRINT SPACE IF POSITIVE
2752 002006' 001000 000000
2753 002007' 000000 000000
2754 002008' 001000 000000
2755 002009' 001000 000000
2756 002010' 001000 000000
2757 002011' 000000 000000
2758

```

```

2752 0020200 000000 002000f          46250   MVI  M,"-"
2753 002015f 001000 000056          46250   MVI  M,"-"
2754 002014f 001000 000055          46280   FOUT1: INX  M
2755 002015f 001000 000043          46280   FOUT1: INX  M
2756 002016f 001000 000066          46300   MVI  M,"0"
2757 002017f 001000 000068          46300   MVI  M,"0"
2758 002018f 001000 000062          46320   JZ   FOUT19
2759 002021f 001000 000064f
2760 002022f 001000 000069          46320   JZ   FOUT19
2761 002023f 001000 000345          46340   PUSH  H
2762 002024f 001000 000374          46360   CM   NEG
2763 002025f 001000 001175f
2764 002026f 001000 002261f
2765

2766 002027f 001000 000257          46400   ;MEHME WE GET THE FAC IN THE RANGE 1000000 .LE, FAC .LE, 999999 AND ROUND IT TO
2767 002028f 001000 000258          46420   ;AN INTEGER, WE KEEP A COUNT OF HOW MANY TIMES WE MULTIPLY OR DIVIDE BY TEN
2768 002029f 001000 000259          46440   ;SO WE KNOW WHAT THE EXPONENT WILL BE. THE FAC IS THEN CONVERTED TO AN
2769 002030f 001000 000260          46460   ;INTEGER IN C,D,E. WE USE A TABLE OF POWERS OF TEN TO CALCULATE EACH DIGIT,
2770 002031f 001000 000261          46480   ;THIS ALGORITHM IS USED FOR SPEED.
2771 002027f 001000 000257          46500   XRA  A           ;PUT TEN'S EXPONENT COUNT ON STACK
2772 002030f 001000 000265          46520   PUSH  PSW
2773 002031f 001000 000319          46540   CALL  FOUTCB
2774 002032f 001000 000314f
2775 002033f 001000 000262f
2776 002034f 001000 000001          46560   FOUT3: MOVR1 221,103,117,370 ;IS NUMBER .LE, 99999,9499? IT IS TOO SMALL
2777 002035f 001000 000103          46580   CALL  FCOMP
2778 002036f 001000 000221
2779 002037f 001000 000221
2780 002040f 001000 000370
2781 002041f 001000 000117
2782 002042f 001000 000315          46580   CALL  FCOMP
2783 002043f 001000 001517f
2784 002044f 001000 002632f
2785
2786 002045f 001000 000342          46600   ;PARITY WILL BE ODD IFF 1 IS RETURNED
2787 002046f 001000 002671f
2788 002047f 001000 002643f
2789 002050f 001000 000561          46640   PUP  PSW
2790 002051f 001000 000561          46660   CALL  FINHLT
2791 002052f 001000 001673f
2792 002053f 001000 002646f
2793 002054f 001000 000365          46680   PUSH  PSW
2794 002055f 001000 000303          46700   JMP  FUUIT3
2795 002056f 001000 002634f
2796 002057f 001000 002652f
2797 002058f 001000 000315          46720   FOUT9: CALL  DIV10
2798 002059f 001000 000317f
2799 002060f 001000 002655f
2800 002063f 001000 000351          46740   POP  PSW
2801 002064f 001000 000674          46760   INR  A
2802 002065f 001000 000365          46780   PUSH  PSW
2803 002066f 001000 000313          46800   CALL  FOUTCB
2804 002067f 001000 002274f

```

```

2805 002070f 001000 002611f
2806
2807
2808
2809 002071f 001000 000315          46820   ;YES, NUMBER IS IN PRINTING RANGE, I,E,
2810 002072f 001000 000000f          46840   ;ALL DIGITS TO BE PRINTED ARE THE INTEGER PART
2811 002073f 001000 002667f
2812 002074f 001000 000674          46860   FOUT5: CALL  FAODH
2813
2814
2815
2816
2817 002075f 001000 000315          46880   INR  A
2818 002076f 001000 001372f
2819 002077f 001000 002676f
2820 002100f 001000 000315          46980   CALL  MOVFR
2821 002101f 001000 001225f
2822 002102f 001000 002676f
2823
2824 002103f 001000 000001          47000   ;DECIDE IF THE NUMBER SHOULD BE PRINTED IN FIXED OR FLOATING NOTATION
2825 002104f 001000 001100*          47000   LXI  B,2*400+6$CODE ;SET DECIMAL POINT COUNT FOR E NOTATION
2826 002105f 001000 002181f
2827 002106f 001000 000051          47040   JC  = DIGIT COUNT
2828
2829 002107f 001000 000201          47060   POP  PSW
2830 002111f 001000 000374          47060   ADD  C
2831 002111f 001000 000374f
2832 002112f 001000 002100f
2833 002113f 001000 002100f
2834 002114f 001000 000007
2835 002115f 001000 000222          47140   JNC  FOUT6
2836 002116f 001000 002124f
2837 002117f 001000 002111f
2838 002118f 001000 000764          47160   INR  A
2839 002119f 001000 000807          47160   MOV  B,A
2840 002122f 001000 000674          47200   MVI  A,1
2841 002123f 001000 000681          47220   ;IF WE ARE USING FIXED POINT NOTATION
2842
2843 002124f 001000 000675          47240   FOUT6: DCR  A
2844 002125f 001000 000341          47240   DCR  A
2845 002126f 001000 000365          47260   PUSH  PSW
2846 002127f 001000 000821          47300   ;CALCULATE THE DIGITS OF THE NUMBER
2847 002128f 001000 000821          47320   LXI  D,FUUTBL
2848 002130f 001000 002310f
2849 002131f 001000 002118f
2850 002132f 001000 000625          47340   FOUT8: DCR  B
2851 002133f 001000 000666          47360   MVI  B,"."
2852 002134f 001000 000565
2853 002135f 001000 000314
2854 002136f 001000 001254f
2855 002137f 001000 002136f
2856 002140f 001000 000685          47480   PUSH  B
2857 002141f 001000 000345          47480   PUSH  H

```

```

F4   MAC 23-AUG-64 06108          FLOATING POINT OUTPUT ROUTINE

2858 002142# 001000 000325    47440  PUSH  D      ;SAVE POWER OF TEN POINTER
2859 002143# 001000 000319    47460  CALL  MOVHF  ;GET NUMBER IN C/D/E
2860 002144# 001000 001249#    47480  POP   H      ;GET POWER OF TEN POINTER
2861 002145# 001000 000202    47500  MVI  B,"0H"-1 ;IB = NEXT DIGIT TO BE PRINTED
2862 002145# 001000 000341    47520  FOUT12: INR  B      ;ADD ONE TO DIGIT
2863 002147# 001000 000006    47540  MOV   A,E    ;SUBTRACT LU
2864 002148# 001000 000513    47560  SUB   M,A
2865 002151# 001000 000004    47580  MOV   E,A
2866 002152# 001000 000173    47600  INX   M      ;POINT TO NEXT BYTE OF POWER OF TEN
2867 002153# 001000 000026    47620  MOV   A,D    ;SUBTRACT MU
2868 002154# 001000 000137    47640  SBB   M
2869 002155# 001000 000043    47660  INX   M
2870 002156# 001000 000172    47680  MOV   A,D    ;POINT TO BEGINNING OF POWER OF TEN
2871 002157# 001000 000036    47700  SBB   M
2872 002158# 001000 000127    47720  INX   M
2873 002159# 001000 000143    47740  MOV   A,C    ;SUBTRACT MO
2874 002162# 001000 000171    47760  SBB   M
2875 002163# 001000 000236    47780  INX   M
2876 002164# 001000 000117    47800  MOV   C,A
2877 002165# 001000 000053    47820  DCX   H
2878 002166# 001000 000053    47840  INX   M      ;INCREMENT POINTER TO NEXT POWER OF TEN
2879 002167# 001000 000322    47860  CALL  FOUT10 ;SUBTRACT AGAIN IF RESULT WAS POSITIVE
2880 002168# 001000 000151    47880  JNC   FOUT10
2881 002171# 001000 002144#    47900  CALL  FADDA ;FIT WASN'T, ADD POWER OF TEN BACK IN
2882 002172# 001000 000815    47920  FADDA
2883 002173# 001000 000274#    47940  INX   M      ;INCREMENT BUFFER POINTER
2884 002174# 001000 0002170#    47960  MOV   B,M
2885 002175# 001000 000043    47980  POP   B      ;GET COUNTERS OFF STACK
2886 002176# 001000 000815    48000  DCR   C
2887 002177# 001000 000253#    48020  INX   M      ;HAS THAT THE LAST DIGIT?
2888 002178# 001000 000211#    48040  JZ    FOUT12 ;DO MORE IF NOT
2889 002201# 001000 000853    48060  XCHG  D,B
2890 002202# 001000 000854    48080  POP   M      ;GET POWER OF TEN POINTER IN (DE)
2891 002203# 001000 000160    48100  MOV   M,B
2892 002204# 001000 000043    48120  CPI   "#"
2893 002205# 001000 000801    48140  INX   M      ;PUT CHARACTER IN BUFFER
2894 002206# 001000 000015    48160  POP   B
2895 002207# 001000 000802    48180  DCR   C      ;GET COUNTERS OFF STACK
2896 002208# 001000 000211#    48200  INX   M      ;HAS THAT THE LAST DIGIT?
2897 002211# 001000 002177#    48220  JZ    FOUT12 ;FIT DOES, WE HAVE NO ZEROS TO SUPPRESS
2898 002212# 001000 000005    48240  DCR   B
2899 002213# 001000 0008512   48260  INX   M      ;SEE IF DECIMAL POINT GOES AFTER LAST DIGIT
2900 002214# 001000 0002232#    48280  JZ    FOUT12
2901 002215# 001000 0002210#    48300  FOUT11: DCR  M      ;SUPPRESS THE TRAILING ZEROS
2902 002216# 001000 000005    48320  INX   M      ;GO BACK TO LAST CHARACTER
2903 002217# 001000 0008176   48340  MOV   A,M
2904 002220# 001000 0008576   48360  CPI   "#"
2905 002221# 001000 0002229#    48380  INX   M      ;IGNORE TRAILING ZEROS
2906 002222# 001000 0000050    48400  CPI   "#"
2907 002222# 001000 0008312   48420  INX   M
2908 002223# 001000 0002218#    48440  JZ    FOUT11
2909 002224# 001000 0002214#    48460  INX   M      ;SUPPRESS DECIMAL POINT IF WE HAVE AN INTEGER
2910

```

```

F4   MAC 23-AUG-64 06108          FLOATING POINT OUTPUT ROUTINE

2911 002225# 001000 000376    48480  CPI   "#,"    ;IGNORE DECIMAL POINT BEFORE TRAILING ZEROS
2912 002226# 001000 0000056   48500  CNZ   INXHRT  ;IF NO DP, MOVE POINTER TO NEXT POSITION
2913 002226# 001000 0008056   48520  INX   M
2914 002227# 001000 0008054   48540  CPI   "#,"    ;PUT SIGN OF EXPONENT IN BUFFER
2915 002221# 001000 0002229#    48560  INP   M      ;INPUT IN THE SIGN OF THE EXPONENT
2916 002232# 001000 0008561   48580  MVI  M,"+"  ;A PLUS IF POSITIVE
2917 002233# 001000 000312    48600  INR   A
2918 002234# 001000 0002271#    48620  FOUT12: POP  PSH
2919 002235# 001000 0000000   48640  JZ    FOUT17 ;GET DECIMAL EXPONENT
2920
2921 002236# 001000 0008065   48660  MVI  M,"E"  ;RETURN IF NUMBER WAS IN FIXED POINT FORMAT
2922 002237# 001000 0008105   48680  CPI   "#,"    ;PUT "E" IN THE BUFFER
2923 002240# 001000 0008043   48700  INX   M
2924
2925 002241# 001000 0000066   48720  INP   M      ;PUT SIGN OF EXPONENT IN BUFFER
2926 002242# 001000 0000053   48740  MVI  M,"+"  ;A PLUS IF POSITIVE
2927 002243# 001000 0003062   48760  INR   A
2928 002244# 001000 0002252#    48780  JP    FOUT14
2929
2930 002245# 001000 0008066   48800  MVI  M,"-"  ;FLOATING POINT NOTATION -- PUT AN "-" IN THE BUFFER
2931 002246# 001000 0008055   48820  CPI   "#,"    ;PUT AN "E" IN THE BUFFER
2932 002247# 001000 0008055   48840  INX   M
2933 002251# 001000 0000074   48860  INR   A
2934
2935 002252# 001000 0000006   48880  FOUT14: MVI  B,"0H"-1 ;CALCULATE TIME TWO DIGIT EXPONENT
2936 002253# 001000 0000057   48900  INR   B      ;INITIALIZE TEN'S DIGIT COUNT
2937 002254# 001000 0000004   48920  FOUT15: INR  B
2938 002255# 001000 0000033   48940  SUI  12     ;INCREMENT DIGIT
2939 002256# 001000 0000112   48960  JNC   FOUT15 ;SUBTRACT TEN
2940 002257# 001000 0008322   48980  INX   M
2941 002258# 001000 0002254#    49000  JZ    FOUT15 ;DO IT AGAIN IF RESULT WAS POSITIVE
2942 002259# 001000 0002244#    49020  INR   A
2943 002252# 001000 0008056   49040  ADI  "#0H+12 ;ADD BACK IN TEN AND CONVERT TO ASCII
2944 002253# 001000 0008072   49060  CPI   "#,"    ;PUT THE EXPONENT IN THE BUFFER
2945
2946 002254# 001000 0008043   49080  INX   M
2947 002255# 001000 0001610   49100  MOV   M,B
2948 002256# 001000 0008043   49120  FOUT19: INX  M      ;PUT TEN'S DIGIT OF EXPONENT IN BUFFER
2949 002257# 001000 0001617   49140  MOV   M,A
2950 002258# 001000 0008043   49160  INX   M      ;WHEN WE JUMP TO HERE, A IS ZERO
2951 002259# 001000 0001611   49180  MOV   C,A      ;PUT ONE'S DIGIT IN BUFFER
2952 002262# 001000 00080341  49200  INX   M      ;INCREMENT POINTER
2953 002263# 001000 00080311  49220  POP   H
2954
2955 002274# 001000 0008681   49240  INX   M      ;PUT ZERO AT END OF BUFFER
2956 002275# 001000 0008164   49260  RET
2957 002276# 001000 0008024   49280  FOUTCB: MOVR1 224,154,043,387 ;COMPARE NUMBER WITH CONSTANT
2958 002277# 001000 00080621  49300  CALL  FCMPB
2959 002277# 001000 00080621  49320
2960 002308# 001000 00080567  49340
2961 002309# 001000 00080435  49360
2962 002302# 001000 00080315  49380
2963 002303# 001000 00081517  49400

```

F4 MAC 23-AUG-64 06108
FLOATING POINT OUTPUT ROUTINE

```

2964 002304F 000000 002260F
2965 002305F 001000 000341      48828    POP    M      IGET RETURN ADDRESS OFF STACK
2966 002306F 001000 000342      48840    JPO    FOUT9    FNNUMBER TOO BIG, DIVIDE BY TEN
2967 002307F 000000 002660F
2968 002310F 000000 002305F
2969 002311F 001000 000351      48850    PCHL    INUMBER OK, RETURN
2970
2971
2972 002312F 000000 000000      48900    ICONSTANTS FOR FOUT
2973 002313F 000000 000000      48920    FHALF: 000 1/2
2974 002314F 000000 000000      48940    002 7TMIS CONSTANT IS ALSO USED BY SQR, SIN, COS
2975 002315F 000000 000000      48960    000
2976
2977 002316F 000000 000240      48980    IPOWER OF TEN TABLE
2978 002317F 000000 000206      48990    FOUTBL: 240 1 100000
2979 002318F 000000 000001      49000    200
2980 002319F 000000 000020      49010    001
2981 002320F 000000 000047      49020    020 1 10000
2982 002322F 000000 000000      49100    047
2983 002324F 000000 000350      49120    000
2984 002325F 000000 000003      49140    350 1 1000
2985 002326F 000000 000000      49160    003
2986 002327F 000000 000144      49200    000
2987 002328F 000000 000000      49220    144 1 1000
2988 002329F 000000 000000      49240    000
2989 00232DF 000000 000012      49260    012 1 10
2990 002333F 000000 000000      49280    000
2991 002334F 000000 000000      49300    000
2992 002335F 000000 000001      49320    001 1
2993 002336F 000000 000000      49340    000
2994 002337F 000000 000000      49360    000
2995
2996 002338F LENGTH=2,4
2997 00400 000000 000000      49400    FPUTPUT THE VALUE IN THE FAC ACCORDING TO THE FORMAT SPECIFICATIONS
2998 00420 000000 000000      49420    I IN A,B,C
2999 00440 000000 000000      49440    FALL REGISTERS ARE ALTERED
2999 00460 000000 000000      49460    ITHE ORIGINAL CONTENTS OF THE FAC IS LOST
3000
3001 00480 000000 000000      49480    ITHE FORMAT IS SPECIFIED IN A, B AND C AS FOLLOWS:
3002 00490 000000 000000      49500    ITHE BITS OF A MEAN THE FOLLOWING:
3003 00490 000000 000000      49540    IBIT 7 0 MEANS FREE FORMAT OUTPUT, I.E. THE OTHER BITS OF A MUST BE ZERO,
3004 00490 000000 000000      49560    TRAILING ZEROS ARE SUPPRESSED, I.E. A NUMBER IS PRINTED IN FIXED OR FLOATING
3005
3006 004980 000000 000003      49580    IPOINT NOTATION ACCORDING TO ITS MAGNITUDE, THE NUMBER IS LEFT
3007 004980 000000 000000      49600    IJUSTIFIED IN ITS FIELD, B AND C ARE IGNORED,
3008 004980 000000 000000      49620    IHEANS FIXE FORMAT OUTPUT, I.E., THE OTHER BITS OF A ARE CHECKED FOR
3009 004980 000000 000000      49640    FORMATTING INFORMATION, THE NUMBER IS RIGHT JUSTIFIED IN ITS FIELD,
3010 004980 000000 000000      49660    ITRAILING ZEROS ARE NOT SUPPRESSED. THIS IS USED FOR PRINTING
3011 004980 000000 000000      49680    I MEANS GROUP THE DIGITS IN THE INTEGER PART OF THE NUMBER INTO GROUPS
3012 004980 000000 000000      49700    IOF THREE AND SEPARATE THE GROUPS BY COMMAS
3013 004980 000000 000000      49720    I 0 MEANS DONT' PRINT THE NUMBER WITH COMMAS
3014 004980 000000 000000      49740    IBIT 5 1 MEANS FILL THE LEADING SPACES IN THE FIELD WITH ASTERisks ("*")
3015 004980 000000 000000      49760    IBIT 4 1 MEANS OUTPUT THE NUMBER WITH A FLOATING DOLLAR SIGN ("$")
3016 004980 000000 000000      49780    IBIT 3 1 MEANS PRINT THE SIGN OF A POSITIVE NUMBER AS A PLUS SIGN ("+")

```

F4 MAC 23-AUG-64 06108
FLOATING POINT OUTPUT ROUTINE

```

3017 004980 000000 000000      49800    IINSTEAD OF A SPACE
3018 004980 000000 000000      49820    IBIT 2 1 MEANS PRINT THE SIGN OF THE NUMBER AFTER THE NUMBER
3019 004980 000000 000000      49840    IBIT 1 UNUSED
3020 004980 000000 000000      49850    I 0 MEANS PRINT THE NUMBER IN FLOATING POINT NOTATION I.E., "E" NOTATION"
3021 004980 000000 000000      49880    I IF THIS BIT IS ON, THE COMMA SPECIFICATION (BIT 6) IS IGNORED,
3022 004980 000000 000000      49900    0 MEANS PRINT THE NUMBER IN FIXED POINT NOTATION, NUMBER .GE, 1E16
3023 004980 000000 000000      49920    I CANNOT BE PRINTED IN FIXED POINT NOTATION,
3024 004980 000000 000000      49940
3025 004980 000000 000000      49960    IB AND C TELL HOW BIG THE FIELD IS!
3026 004980 000000 000000      49980    IB = THE NUMBER OF PLACES IN THE FIELD TO THE LEFT OF THE DECIMAL POINT
3027 004980 000000 000000      50000    IC = (B+C) INCLUDES THE DECIMAL POINT)
3028 004980 000000 000000      50020    IC = THE NUMBER OF PLACES IN THE FIELD TO THE RIGHT OF THE DECIMAL POINT
3029 004980 000000 000000      50040    (C INCLUDES THE DECIMAL POINT)
3030 004980 000000 000000      50060    B AND C DONT INCLUDE THE 4 POSITIONS FOR THE EXPONENT IF BIT 0 IS ON
3031 004980 000000 000000      50080    FFOUT ASSUMES B+C ,LE, 24 (DECIMAL)
3032 004980 000000 000000      50100
3033 004980 000000 000000      50120    FOUT1: FTRY TO PRINT THE FAC IN FREE FORMAT
3034 004980 000000 000000      50140    XRA 00000000000000000000000000000000    ISET FORMAT FLAGS TO FREE FORMATED OUTPUT
3035 004980 000000 000000      50160    FPUT1: FTRY TO PRINT THE FAC USING THE SPECIFICATIONS IN A AND C
3036 004980 000000 000000      50180    PUFOUT1: CALL FOUINI    ISAVE THE FORMAT SPECIFICATION IN A AND C
3037 004980 000000 000000      50200
3038 004980 000000 000000      50220    PUSH B    ISAVE THE FIELD LENGTH SPECIFICATIONS
3039 004980 000000 000000      50240    ANI 10    ICHECK IF POSITIVE NUMBERS GET A PLUS SIGN
3040 004980 000000 000000      50260    JZ FOUT1    ITHEY DON'T
3041 004980 000000 000000      50280    MVI M,"+"    ITHEY DO, PUT IN A PLUS SIGN
3042 004980 000000 000000      50300    FOUT1: LAH TEMP1    ISSEE WHAT KIND OF A VALUE WE HAVE
3043 004980 000000 000000      50320    MOV B,A    ISAVE IT
3044 004980 000000 000000      50340    XCHG    ISAVE BUFFER POINTER
3045 004980 000000 000000      50360    CALL VSIGN    IGET THE SIGN OF THE FAC
3046 004980 000000 000000      50380    XCHG    IPUT THE BUFFER POINTER BACK IN (HL)
3047 004980 000000 000000      50400    MOV A,B    IGET THE VALTYP BACK
3048 004980 000000 000000      50420    JP FOUT2    IF WE HAVE A NEGATIVE NUMBER, NEGATE IT
3049 004980 000000 000000      50440    MVI M,"-"    I AND PUT A MINUS SIGN IN THE BUFFER
3050 004980 000000 000000      50460    PUSH M    ISAVE THE BUFFER POINTER
3051 004980 000000 000000      50480    CALL VNNEG    INVERSE THE NUMBER
3052 004980 000000 000000      50500    POP M    IGET THE BUFFER POINTER BACK
3053 004980 000000 000000      50520    FOUT2: INX M    IPPOINT TO WHERE THE NEXT CHARACTER GOES
3054 004980 000000 000000      50540    LDA TEMP3    IGET THE FORMAT SPECIFICATION
3055 004980 000000 000000      50560    MOV D,A    ISAVE IT FOR LATER
3056 004980 000000 000000      50580    RAL    IPUT THE FREE FORMATED OR NOT BIT IN THE CARRY
3057 004980 000000 000000      50600    LDA VALTYP    IGET THE VALTYP, VNNEG COULD HAVE CHANGED THIS
3058 004980 000000 000000      50620    ; SINCE THE 32765 IS INT AND 32768 IS SNG,
3059 004980 000000 000000      50640    ; SNG IS NOT EQUIVALENT
3060 004980 000000 000000      50660    JC FOUTFX    ITHE MAN WANTS FIXED FORMATED OUTPUT
3061 004980 000000 000000      50680    IHERE TO PRINT NUMBERS IN FREE FORMAT
3062 004980 000000 000000      50700    POP B    IME CAN IGNORE THE OLD B AND C
3063 004980 000000 000000      50720    MVI M,"0"    IPUT A ZERO IN THE BUFFER IN CASE THE NUMBER
3064 004980 000000 000000      50740    JZ FOUTZR    I IS ZERO, IT IS, FINISH IT UP
3065 004980 000000 000000      50760    CPI 4    IDECIDE WHAT KIND OF A VALUE WE HAVE
3066 004980 000000 000000      50780    JNC FOUPFRV    IME HAVE A SNG OR DBL
3067 004980 000000 000000      50800    IHERE TO PRINT AN INTEGER IN FREE FORMAT
3068 004980 000000 000000      50820    LXI B,SCODE    ISET THE DECIMAL POINT COUNT AND COMMA COUNT
3069 004980 000000 000000      50840    ; TO ZERO

```

```

3078      S1880    CALL    FOUTCI   ;CONVERT THE INTEGER TO DECIMAL
3079      S1881    MOV     A,B   ;IF ALL INTO FOUTZS AND ZERO SUPPRESS THE THING
3080      S1882
3081      S1883    FOUTZS1  LXI    H,FBUFR+1 ;ZERO SUPPRESS THE DIGITS IN FBUFFR
3082      S1884    MVI    C,0   ;FASTERISK FILL AND ZERO SUPPRESS IF NECESSARY
3083      S1885    JSET UP B AND CONDITION CODES IF WE HAVE A.TRAILING SIGN
3084      S1886    FOUTZS1  LXI    H,FBUFR+1 ;GET PRINTED TO THE SIGN
3085      S1887    MOV     B,F   ;SAVE THE SIGN TO THE SIGN
3086      S1888    MVI    C," " ;DEFAULT FILL CHARACTER TO A SPACE
3087      S1889    LDA    TEMP3  ;GET FORMAT SPECS TO SEE IF WE HAVE TO
3088      S1890    MOV     E,A   ;ASTERISK FILL, SAVE IT
3089      S1891    ANI    #40
3090      S1892    JZ     FOUTZS1 ;WE DON'T
3091      S1893    MOV     A,B   ;WE DO, SEE IF THE SIGN WAS A SPACE
3092      S1894    CMP     C," "
3093      S1895    MVI    C," " ;ZERO FLAG IS SET, IT WAS
3094      S1896    JZ     FOUTZS1 ;SET THE FILL CHARACTER TO AN ASTERISK
3095      S1897    MOV     B,C   ;SET THE SIGN TO AN ASTERISK IF IT HAS A SPACE
3096      S1898    MVI    C," "
3097      S1899    JZ     FOUTZS1 ;JB HAS THE SIGN, C THE FILL CHARACTER
3098      S1900    MOV     M,C   ;FILL IN THE ZERO OR THE SIGN
3099      S1901    CPI    "#"
3100      S1902    JZ     FOUTZS1 ;WE GET THE NEXT CHARACTER IN THE BUFFER
3101      S1903    CPI    ","
3102      S1904    JZ     FOUTZS1 ;IF THERE ARE NO SPACES, "CHRGET" IS
3103      S1905    CPI    "#"
3104      S1906    JZ     FOUTZS1 ;EQUIVALENT TO "INX H//MOV A,M"
3105      S1907    MOV     A,B   ;DO WE HAVE A ZERO?
3106      S1908    CPI    "#"
3107      S1909    JZ     FOUTZS1 ;YES, SUPPRESS IT
3108      S1910    MVI    C," "
3109      S1911    JZ     FOUTZS1 ;ISAN WE DO WE HAVE A COMMA?
3110      S1911    CPI    ","
3111      S1912    JZ     FOUTZS1 ;YES, SUPPRESS IT
3112      S1913    CPI    "#"
3113      S1914    JZ     FOUTZS1 ;ARE WE AT THE DECIMAL POINT?
3114      S1915    CPI    "#"
3115      S1916    JZ     FOUTZS1 ;NO, I GUESS NOT
3116      S1917    DCX    H   ;YES, BACK UP AND PUT A ZERO BEFORE IT
3117      S1918    MVI    M," "
3118      S1919    FOUTZS1  MOV     A,E   ;GET THE FORMAT SPECS TO CHECK FOR A FLOATING
3119      S1920    CPI    "#"
3120      S1921    JZ     FOUTZS1 ;DOLLAR SIGN
3121      S1922    DCX    H   ;WE DON'T HAVE ONE
3122      S1923    MVI    C," "
3123      S1924    FOUTZS1  MOV     A,E   ;WE HAVE ONE, BACK UP AND PUT IN THE DOLLAR
3124      S1925    CPI    "#"
3125      S1926    JZ     FOUTZS1 ;SIGN
3126      S1927    DCX    H   ;DO WE HAVE A TRAILING SIGN?
3127      S1928    MVI    C," "
3128      S1929    FOUTZS1  MOV     A,E   ;HERE TO INITIALLY SET UP THE FORMAT SPECS AND PUT IN A SPACE FOR THE
3129      S1930    CPI    "#"
3130      S1931    FOUIN1  STA    TEMP3  ;SIGN OF A POSITIVE NUMBER
3131      S1932    MVI    C," "
3132      S1933    FOUIN1  LXI    H,FBUFR+1 ;SAVE THE FORMAT SPECIFICATION
3133      S1934    MVI    M," "
3134      S1935    FOUIN1  RET
3135      S1936    CPI    "#"
3136      S1937    FOUIN1  RET
3137      S1938    CPI    "#"
3138      S1939    FOUIN1  JNC    FOUTNV ;PUT A POINTER INTO FBUFFR
3139      S1940    MVI    C," "
3140      S1941    FOUIN1  RET
3141      S1942    CPI    "#"
3142      S1943    FOUIN1  RET
3143      S1944    CPI    "#"
3144      S1945    FOUIN1  RET
3145      S1946    CPI    "#"
3146      S1947    FOUIN1  RET
3147      S1948    CPI    "#"
3148      S1949    FOUIN1  RET
3149      S1950    CPI    "#"
3150      S1951    FOUIN1  RET
3151      S1952    CPI    "#"
3152      S1953    FOUIN1  RET
3153      S1954    CPI    "#"
3154      S1955    FOUIN1  RET
3155      S1956    CPI    "#"
3156      S1957    FOUIN1  RET
3157      S1958    CPI    "#"
3158      S1959    FOUIN1  RET
3159      S1960    CPI    "#"
3160      S1961    FOUIN1  RET
3161      S1962    CPI    "#"
3162      S1963    FOUIN1  RET
3163      S1964    CPI    "#"
3164      S1965    FOUIN1  RET
3165      S1966    CPI    "#"
3166      S1967    FOUIN1  RET
3167      S1968    CPI    "#"
3168      S1969    FOUIN1  RET
3169      S1970    CPI    "#"
3170      S1971    FOUIN1  RET
3171      S1972    CPI    "#"
3172      S1973    FOUIN1  RET
3173      S1974    CPI    "#"
3174      S1975    FOUIN1  RET
3175      S1976    CPI    "#"
3176      S1977    FOUIN1  RET
3177      S1978    CPI    "#"
3178      S1979    FOUIN1  RET
3179      S1980    CPI    "#"
3180      S1981    FOUIN1  RET
3181      S1982    CPI    "#"
3182      S1983    FOUIN1  RET
3183      S1984    CPI    "#"
3184      S1985    FOUIN1  RET
3185      S1986    CPI    "#"
3186      S1987    FOUIN1  RET
3187      S1988    CPI    "#"
3188      S1989    FOUIN1  RET
3189      S1990    CPI    "#"
3190      S1991    FOUIN1  RET
3191      S1992    CPI    "#"
3192      S1993    FOUIN1  RET
3193      S1994    CPI    "#"
3194      S1995    FOUIN1  RET
3195      S1996    CPI    "#"
3196      S1997    FOUIN1  RET
3197      S1998    CPI    "#"
3198      S1999    FOUIN1  RET
3199      S1900    CPI    "#"
3200      S1901    FOUIN1  RET
3201      S1902    CPI    "#"
3202      S1903    FOUIN1  RET
3203      S1904    CPI    "#"
3204      S1905    FOUIN1  RET
3205      S1906    CPI    "#"
3206      S1907    FOUIN1  RET
3207      S1908    CPI    "#"
3208      S1909    FOUIN1  RET
3209      S1910    CPI    "#"
3210      S1911    FOUIN1  RET
3211      S1912    CPI    "#"
3212      S1913    FOUIN1  RET
3213      S1914    CPI    "#"
3214      S1915    FOUIN1  RET
3215      S1916    CPI    "#"
3216      S1917    FOUIN1  RET
3217      S1918    CPI    "#"
3218      S1919    FOUIN1  RET
3219      S1920    CPI    "#"
3220      S1921    FOUIN1  RET
3221      S1922    CPI    "#"
3222      S1923    FOUIN1  RET
3223      S1924    CPI    "#"
3224      S1925    FOUIN1  RET
3225      S1926    CPI    "#"
3226      S1927    FOUIN1  RET
3227      S1928    CPI    "#"
3228      S1929    FOUIN1  RET
3229      S1930    CPI    "#"
3230      S1931    FOUIN1  RET
3231      S1932    CPI    "#"
3232      S1933    FOUIN1  RET
3233      S1934    CPI    "#"
3234      S1935    FOUIN1  RET
3235      S1936    CPI    "#"
3236      S1937    FOUIN1  RET
3237      S1938    CPI    "#"
3238      S1939    FOUIN1  RET
3239      S1940    CPI    "#"
3240      S1941    FOUIN1  RET
3241      S1942    CPI    "#"
3242      S1943    FOUIN1  RET
3243      S1944    CPI    "#"
3244      S1945    FOUIN1  RET
3245      S1946    CPI    "#"
3246      S1947    FOUIN1  RET
3247      S1948    CPI    "#"
3248      S1949    FOUIN1  RET
3249      S1950    CPI    "#"
3250      S1951    FOUIN1  RET
3251      S1952    CPI    "#"
3252      S1953    FOUIN1  RET
3253      S1954    CPI    "#"
3254      S1955    FOUIN1  RET
3255      S1956    CPI    "#"
3256      S1957    FOUIN1  RET
3257      S1958    CPI    "#"
3258      S1959    FOUIN1  RET
3259      S1960    CPI    "#"
3260      S1961    FOUIN1  RET
3261      S1962    CPI    "#"
3262      S1963    FOUIN1  RET
3263      S1964    CPI    "#"
3264      S1965    FOUIN1  RET
3265      S1966    CPI    "#"
3266      S1967    FOUIN1  RET
3267      S1968    CPI    "#"
3268      S1969    FOUIN1  RET
3269      S1970    CPI    "#"
3270      S1971    FOUIN1  RET
3271      S1972    CPI    "#"
3272      S1973    FOUIN1  RET
3273      S1974    CPI    "#"
3274      S1975    FOUIN1  RET
3275      S1976    CPI    "#"
3276      S1977    FOUIN1  RET
3277      S1978    CPI    "#"
3278      S1979    FOUIN1  RET
3279      S1980    CPI    "#"
3280      S1981    FOUIN1  RET
3281      S1982    CPI    "#"
3282      S1983    FOUIN1  RET
3283      S1984    CPI    "#"
3284      S1985    FOUIN1  RET
3285      S1986    CPI    "#"
3286      S1987    FOUIN1  RET
3287      S1988    CPI    "#"
3288      S1989    FOUIN1  RET
3289      S1990    CPI    "#"
3290      S1991    FOUIN1  RET
3291      S1992    CPI    "#"
3292      S1993    FOUIN1  RET
3293      S1994    CPI    "#"
3294      S1995    FOUIN1  RET
3295      S1996    CPI    "#"
3296      S1997    FOUIN1  RET
3297      S1998    CPI    "#"
3298      S1999    FOUIN1  RET
3299      S1900    CPI    "#"
3300      S1901    FOUIN1  RET
3301      S1902    CPI    "#"
3302      S1903    FOUIN1  RET
3303      S1904    CPI    "#"
3304      S1905    FOUIN1  RET
3305      S1906    CPI    "#"
3306      S1907    FOUIN1  RET
3307      S1908    CPI    "#"
3308      S1909    FOUIN1  RET
3309      S1910    CPI    "#"
3310      S1911    FOUIN1  RET
3311      S1912    CPI    "#"
3312      S1913    FOUIN1  RET
3313      S1914    CPI    "#"
3314      S1915    FOUIN1  RET
3315      S1916    CPI    "#"
3316      S1917    FOUIN1  RET
3317      S1918    CPI    "#"
3318      S1919    FOUIN1  RET
3319      S1920    CPI    "#"
3320      S1921    FOUIN1  RET
3321      S1922    CPI    "#"
3322      S1923    FOUIN1  RET
3323      S1924    CPI    "#"
3324      S1925    FOUIN1  RET
3325      S1926    CPI    "#"
3326      S1927    FOUIN1  RET
3327      S1928    CPI    "#"
3328      S1929    FOUIN1  RET
3329      S1930    CPI    "#"
3330      S1931    FOUIN1  RET
3331      S1932    CPI    "#"
3332      S1933    FOUIN1  RET
3333      S1934    CPI    "#"
3334      S1935    FOUIN1  RET
3335      S1936    CPI    "#"
3336      S1937    FOUIN1  RET
3337      S1938    CPI    "#"
3338      S1939    FOUIN1  RET
3339      S1940    CPI    "#"
3340      S1941    FOUIN1  RET
3341      S1942    CPI    "#"
3342      S1943    FOUIN1  RET
3343      S1944    CPI    "#"
3344      S1945    FOUIN1  RET
3345      S1946    CPI    "#"
3346      S1947    FOUIN1  RET
3347      S1948    CPI    "#"
3348      S1949    FOUIN1  RET
3349      S1950    CPI    "#"
3350      S1951    FOUIN1  RET
3351      S1952    CPI    "#"
3352      S1953    FOUIN1  RET
3353      S1954    CPI    "#"
3354      S1955    FOUIN1  RET
3355      S1956    CPI    "#"
3356      S1957    FOUIN1  RET
3357      S1958    CPI    "#"
3358      S1959    FOUIN1  RET
3359      S1960    CPI    "#"
3360      S1961    FOUIN1  RET
3361      S1962    CPI    "#"
3362      S1963    FOUIN1  RET
3363      S1964    CPI    "#"
3364      S1965    FOUIN1  RET
3365      S1966    CPI    "#"
3366      S1967    FOUIN1  RET
3367      S1968    CPI    "#"
3368      S1969    FOUIN1  RET
3369      S1970    CPI    "#"
3370      S1971    FOUIN1  RET
3371      S1972    CPI    "#"
3372      S1973    FOUIN1  RET
3373      S1974    CPI    "#"
3374      S1975    FOUIN1  RET
3375      S1976    CPI    "#"
3376      S1977    FOUIN1  RET
3377      S1978    CPI    "#"
3378      S1979    FOUIN1  RET
3379      S1980    CPI    "#"
3380      S1981    FOUIN1  RET
3381      S1982    CPI    "#"
3382      S1983    FOUIN1  RET
3383      S1984    CPI    "#"
3384      S1985    FOUIN1  RET
3385      S1986    CPI    "#"
3386      S1987    FOUIN1  RET
3387      S1988    CPI    "#"
3388      S1989    FOUIN1  RET
3389      S1990    CPI    "#"
3390      S1991    FOUIN1  RET
3391      S1992    CPI    "#"
3392      S1993    FOUIN1  RET
3393      S1994    CPI    "#"
3394      S1995    FOUIN1  RET
3395      S1996    CPI    "#"
3396      S1997    FOUIN1  RET
3397      S1998    CPI    "#"
3398      S1999    FOUIN1  RET
3399      S1900    CPI    "#"
3400      S1901    FOUIN1  RET
3401      S1902    CPI    "#"
3402      S1903    FOUIN1  RET
3403      S1904    CPI    "#"
3404      S1905    FOUIN1  RET
3405      S1906    CPI    "#"
3406      S1907    FOUIN1  RET
3407      S1908    CPI    "#"
3408      S1909    FOUIN1  RET
3409      S1910    CPI    "#"
3410      S1911    FOUIN1  RET
3411      S1912    CPI    "#"
3412      S1913    FOUIN1  RET
3413      S1914    CPI    "#"
3414      S1915    FOUIN1  RET
3415      S1916    CPI    "#"
3416      S1917    FOUIN1  RET
3417      S1918    CPI    "#"
3418      S1919    FOUIN1  RET
3419      S1920    CPI    "#"
3420      S1921    FOUIN1  RET
3421      S1922    CPI    "#"
3422      S1923    FOUIN1  RET
3423      S1924    CPI    "#"
3424      S1925    FOUIN1  RET
3425      S1926    CPI    "#"
3426      S1927    FOUIN1  RET
3427      S1928    CPI    "#"
3428      S1929    FOUIN1  RET
3429      S1930    CPI    "#"
3430      S1931    FOUIN1  RET
3431      S1932    CPI    "#"
3432      S1933    FOUIN1  RET
3433      S1934    CPI    "#"
3434      S1935    FOUIN1  RET
3435      S1936    CPI    "#"
3436      S1937    FOUIN1  RET
3437      S1938    CPI    "#"
3438      S1939    FOUIN1  RET
3439      S1940    CPI    "#"
3440      S1941    FOUIN1  RET
3441      S1942    CPI    "#"
3442      S1943    FOUIN1  RET
3443      S1944    CPI    "#"
3444      S1945    FOUIN1  RET
3445      S1946    CPI    "#"
3446      S1947    FOUIN1  RET
3447      S1948    CPI    "#"
3448      S1949    FOUIN1  RET
3449      S1950    CPI    "#"
3450      S1951    FOUIN1  RET
3451      S1952    CPI    "#"
3452      S1953    FOUIN1  RET
3453      S1954    CPI    "#"
3454      S1955    FOUIN1  RET
3455      S1956    CPI    "#"
3456      S1957    FOUIN1  RET
3457      S1958    CPI    "#"
3458      S1959    FOUIN1  RET
3459      S1960    CPI    "#"
3460      S1961    FOUIN1  RET
3461      S1962    CPI    "#"
3462      S1963    FOUIN1  RET
3463      S1964    CPI    "#"
3464      S1965    FOUIN1  RET
3465      S1966    CPI    "#"
3466      S1967    FOUIN1  RET
3467      S1968    CPI    "#"
3468      S1969    FOUIN1  RET
3469      S1970    CPI    "#"
3470      S1971    FOUIN1  RET
3471      S1972    CPI    "#"
3472      S1973    FOUIN1  RET
3473      S1974    CPI    "#"
3474      S1975    FOUIN1  RET
3475      S1976    CPI    "#"
3476      S1977    FOUIN1  RET
3477      S1978    CPI    "#"
3478      S1979    FOUIN1  RET
3479      S1980    CPI    "#"
3480      S1981    FOUIN1  RET
3481      S1982    CPI    "#"
3482      S1983    FOUIN1  RET
3483      S1984    CPI    "#"
3484      S1985    FOUIN1  RET
3485      S1986    CPI    "#"
3486      S1987    FOUIN1  RET
3487      S1988    CPI    "#"
3488      S1989    FOUIN1  RET
3489      S1990    CPI    "#"
3490      S1991    FOUIN1  RET
3491      S1992    CPI    "#"
3492      S1993    FOUIN1  RET
3493      S1994    CPI    "#"
3494      S1995    FOUIN1  RET
3495      S1996    CPI    "#"
3496      S1997    FOUIN1  RET
3497      S1998    CPI    "#"
3498      S1999    FOUIN1  RET
3499      S1900    CPI    "#"
3500      S1901    FOUIN1  RET
3501      S1902    CPI    "#"
3502      S1903    FOUIN1  RET
3503      S1904    CPI    "#"
3504      S1905    FOUIN1  RET
3505      S1906    CPI    "#"
3506      S1907    FOUIN1  RET
3507      S1908    CPI    "#"
3508      S1909    FOUIN1  RET
3509      S1910    CPI    "#"
3510      S1911    FOUIN1  RET
3511      S1912    CPI    "#"
3512      S1913    FOUIN1  RET
3513      S1914    CPI    "#"
3514      S1915    FOUIN1  RET
3515      S1916    CPI    "#"
3516      S1917    FOUIN1  RET
3517      S1918    CPI    "#"
3518      S1919    FOUIN1  RET
3519      S1920    CPI    "#"
3520      S1921    FOUIN1  RET
3521      S1922    CPI    "#"
3522      S1923    FOUIN1  RET
3523      S1924    CPI    "#"
3524      S1925    FOUIN1  RET
3525      S1926    CPI    "#"
3526      S1927    FOUIN1  RET
3527      S1928    CPI    "#"
3528      S1929    FOUIN1  RET
3529      S1930    CPI    "#"
3530      S1931    FOUIN1  RET
3531      S1932    CPI    "#"
3532      S1933    FOUIN1  RET
3533      S1934    CPI    "#"
3534      S1935    FOUIN1  RET
3535      S1936    CPI    "#"
3536      S1937    FOUIN1  RET
3537      S1938    CPI    "#"
3538      S1939    FOUIN1  RET
3539      S1940    CPI    "#"
3540      S1941    FOUIN1  RET
3541      S1942    CPI    "#"
3542      S1943    FOUIN1  RET
3543      S1944    CPI    "#"
3544      S1945    FOUIN1  RET
3545      S1946    CPI    "#"
3546      S1947    FOUIN1  RET
3547      S1948    CPI    "#"
3548      S1949    FOUIN1  RET
3549      S1950    CPI    "#"
3550      S1951    FOUIN1  RET
3551      S1952    CPI    "#"
3552      S1953    FOUIN1  RET
3553      S1954    CPI    "#"
3554      S1955    FOUIN1  RET
3555      S1956    CPI    "#"
3556      S1957    FOUIN1  RET
3557      S1958    CPI    "#"
3558      S1959    FOUIN1  RET
3559      S1960    CPI    "#"
3560      S1961    FOUIN1  RET
3561      S1962    CPI    "#"
3562      S1963    FOUIN1  RET
3563      S1964    CPI    "#"
3564      S1965    FOUIN1  RET
3565      S1966    CPI    "#"
3566      S1967    FOUIN1  RET
3567      S1968    CPI    "#"
3568      S1969    FOUIN1  RET
3569      S1970    CPI    "#"
3570      S1971    FOUIN1  RET
3571      S1972    CPI    "#"
3572      S1973    FOUIN1  RET
3573      S1974    CPI    "#"
3574      S1975    FOUIN1  RET
3575      S1976    CPI    "#"
3576      S1977    FOUIN1  RET
3577      S1978    CPI    "#"
3578      S1979    FOUIN1  RET
3579      S1980    CPI    "#"
3580      S1981    FOUIN1  RET
3581      S1982    CPI    "#"
3582      S1983    FOUIN1  RET
3583      S1984    CPI    "#"
3584      S1985    FOUIN1  RET
3585      S1986    CPI    "#"
3586      S1987    FOUIN1  RET
3587      S1988    CPI    "#"
3588      S1989    FOUIN1  RET
3589      S1990    CPI    "#"
3590      S1991    FOUIN1  RET
3591      S1992    CPI    "#"
3592      S1993    FOUIN1  RET
3593      S1994    CPI    "#"
3594      S1995    FOUIN1  RET
3595      S1996    CPI    "#"
3596      S1997    FOUIN1  RET
3597      S1998    CPI    "#"
3598      S1999    FOUIN1  RET
3599      S1900    CPI    "#"
3600      S1901    FOUIN1  RET
3601      S1902    CPI    "#"
3602      S1903    FOUIN1  RET
3603      S1904    CPI    "#"
3604      S1905    FOUIN1  RET
3605      S1906    CPI    "#"
3606      S1907    FOUIN1  RET
3607      S1908    CPI    "#"
3608      S1909    FOUIN1  RET
3609      S1910    CPI    "#"
3610      S1911    FOUIN1  RET
3611      S1912    CPI    "#"
3612      S1913    FOUIN1  RET
3613      S1914    CPI    "#"
3614      S1915    FOUIN1  RET
3615      S1916    CPI    "#"
3616      S1917    FOUIN1  RET
3617      S1918    CPI    "#"
3618      S1919    FOUIN1  RET
3619      S1920    CPI    "#"
3620      S1921    FOUIN1  RET
3621      S1922    CPI    "#"
3622      S1923    FOUIN1  RET
3623      S1924    CPI    "#"
3624      S1925    FOUIN1  RET
3625      S1926    CPI    "#"
3626      S1927    FOUIN1  RET
3627      S1928    CPI    "#"
3628      S1929    FOUIN1  RET
3629      S1930    CPI    "#"
3630      S1931    FOUIN1  RET
3631      S1932    CPI    "#"
3632      S1933    FOUIN1  RET
3633      S1934    CPI    "#"
3634      S1935    FOUIN1  RET
3635      S1936    CPI    "#"
3636      S1937    FOUIN1  RET
3637      S1938    CPI    "#"
3638      S1939    FOUIN1  RET
3639      S1940    CPI    "#"
3640      S1941    FOUIN1  RET
3641      S1942    CPI    "#"
3642      S1943    FOUIN1  RET
3643      S1944    CPI    "#"
3644      S1945    FOUIN1  RET
3645      S1946    CPI    "#"
3646      S1947    FOUIN1  RET
3647      S1948    CPI    "#"
3648      S1949    FOUIN1  RET
3649      S1950    CPI    "#"
3650      S1951    FOUIN1  RET
3651      S1952    CPI    "#"
3652      S1953    FOUIN1  RET
3653      S1954    CPI    "#"
3654      S1955    FOUIN1  RET
3655      S1956    CPI    "#"
3656      S1957    FOUIN1  RET
3657      S1958    CPI    "#"
3658      S1959    FOUIN1  RET
3659      S1960    CPI    "#"
3660      S1961    FOUIN1  RET
3661      S1962    CPI    "#"
3662      S1963    FOUIN1  RET
3663      S1964    CPI    "#"
3664      S1965    FOUIN1  RET
3665      S1966    CPI    "#"
3666      S1967    FOUIN1  RET
3667      S1968    CPI    "#"
3668      S1969    FOUIN1  RET
3669      S1970    CPI    "#"
3670      S1971    FOUIN1  RET
3671      S1972    CPI    "#"
3672      S1973    FOUIN1  RET
3673      S1974    CPI    "#"
3674      S1975    FOUIN1  RET
3675      S1976    CPI    "#"
3676      S1977    FOUIN1  RET
3677      S1978    CPI    "#"
3678      S1979    FOUIN1  RET
3679      S1980    CPI    "#"
3680      S1981    FOUIN1  RET
3681      S1982    CPI    "#"
3682      S1983    FOUIN1  RET
3683      S1984    CPI    "#"
3684      S1985    FOUIN1  RET
3685      S1986    CPI    "#"
3686      S1987    FOUIN1  RET
3687      S1988    CPI    "#"
3688      S1989    FOUIN1  RET
3689      S1990    CPI    "#"
3690      S1991    FOUIN1  RET
3691      S1992    CPI    "#"
3692      S1993    FOUIN1  RET
3693      S1994    CPI    "#"
3694      S1995    FOUIN1  RET
3695      S1996    CPI    "#"
3696      S1997    FOUIN1  RET
3697      S1998    CPI    "#"
3698      S1999    FOUIN1  RET
3699      S1900    CPI    "#"
3700      S1901    FOUIN1  RET
3701      S1902    CPI    "#"
3702      S1903    FOUIN1  RET
3703      S1904    CPI    "#"
3704      S1905    FOUIN1  RET
3705      S1906    CPI    "#"
3706      S1907    FOUIN1  RET
3707      S1908    CPI    "#"
3708      S1909    FOUIN1  RET
3709      S1910    CPI    "#"
3710      S1911    FOUIN1  RET
3711      S1912    CPI    "#"
3712      S1913    FOUIN1  RET
3713      S1914    CPI    "#"
3714      S1915    FOUIN1  RET
3715      S1916    CPI    "#"
3716      S1917    FOUIN1  RET
3717      S1918    CPI    "#"
3718      S1919    FOUIN1  RET
3719      S1920    CPI    "#"
3720      S1921    FOUIN1  RET
3721      S1922    CPI    "#"
3722      S1923    FOUIN1  RET
3723      S1924    CPI    "#"
3724      S1925    FOUIN1  RET
3725      S1926    CPI    "#"
3726      S1927    FOUIN1  RET
3727      S1928    CPI    "#"
3728      S1929    FOUIN1  RET
3729      S1930    CPI    "#"
3730      S1931    FOUIN1  RET
3731      S1932    CPI    "#"
3732      S1933    FOUIN1  RET
3733      S1934    CPI    "#"
3734      S1935    FOUIN1  RET
3735      S1936    CPI    "#"
3736      S1937    FOUIN1  RET
3737      S1938    CPI    "#"
3738      S1939    FOUIN1  RET
3739      S1940    CPI    "#"
3740      S1941    FOUIN1  RET
3741      S1942    CPI    "#"
3742      S1943    FOUIN1  RET
3743      S1944    CPI    "#"
3744      S1945    FOUIN1  RET
3745      S1946    CPI    "#"
3746      S1947    FOUIN1  RET
3747      S1948    CPI    "#"
3748      S1949    FOUIN1  RET
3749      S1950    CPI    "#"
3750      S1951    FOUIN1  RET
3751      S1952    CPI    "#"
3752      S1953    FOUIN1  RET
3753      S1954    CPI    "#"
3754      S1955    FOUIN1  RET
3755      S1956    CPI    "#"
3756      S1957    FOUIN1  RET
3757      S1958    CPI    "#"
3758      S1959    FOUIN1  RET
3759      S1960    CPI    "#"
3760      S1961    FOUIN1  RET
3761      S1962    CPI    "#"
3762      S1963    FOUIN1  RET
3763      S1964    CPI    "#"
3764      S1965    FOUIN1  RET
3765      S1966    CPI    "#"
3766      S1967    FOUIN1  RET
3767      S1968    CPI    "#"
3768      S1969    FOUIN1  RET
3769      S1970    CPI    "#"
3770      S1971    FOUIN1  RET
3771      S1972    CPI    "#"
3772      S1973    FOUIN1  RET
3773      S1974    CPI    "#"
3774      S1975    FOUIN1  RET
3775      S1976    CPI    "#"
3776      S1977    FOUIN1  RET
3777      S1978    CPI    "#"
3778      S1979    FOUIN1  RET
3779      S1980    CPI    "#"
3780      S1981    FOUIN1  RET
3781      S1982    CPI    "#"
3782      S1983    FOUIN1  RET
3783      S1984    CPI    "#"
3784      S1985    FOUIN1  RET
3785      S1986    CPI    "#"
3786      S1987    FOUIN1  RET
3787      S1988    CPI    "#"
3788      S1989    FOUIN1  RET
3789      S19
```

```

3176      52980    ;CALCULATE THE TWO DIGIT EXPONENT
3177      53000    FOUCCE1: MVI B,"0"=1   ;INITIALIZE TEN'S DIGIT COUNT
3178      53020    FOUCCE2: INR B           ;INCREMENT DIGIT
3179      53040    SUI 12             ;SUBTRACT TEN
3180      53060    JNC FOUCE2        ;DO IT AGAIN IF RESULT WAS POSITIVE
3181      53080    ADI "#P#12         ;ADD BACK IN TEN AND CONVERT TO ASCII
3182      53100    ;PUT THE EXPONENT IN THE BUFFER
3183      53120    INX H             ;INCREMENT BUFFER
3184      53140    MOV M,B           ;PUT TEN'S DIGIT OF EXPONENT IN BUFFER
3185      53160    INX H             ;WHEN WE JUMP TO HERE, A IS ZERO
3186      53180    MOV M,A           ;PUT ONE'S DIGIT IN BUFFER
3187      53200    FOUTZR: INX H          ;INCREMENT POINTER, HERE TO FINISH UP PRINTING
3188      53220    ;A FREE FORMAT ZERO
3189      53240    FOUTON: MVI M,0       ;PUT A ZERO AT THE END OF THE NUMBER
3190      53260    XCHG              ;SAVE THE POINTER AT THE END OF THE NUMBER
3191      53280    ;IN (DE) FOR FFXXFL
3192      53300    LXI H,FBUFFR+1     ;GET A POINTER TO THE BEGINNING
3193      53320    RET               ;ALL DONE
3194
3195
3196      53360    ;HERE TO PUT A POSSIBLE COMMA COUNT IN C, AND ZERO C IF WE ARE NOT
3197      53380    ;USING THE COMMA SPECIFICATION
3198      53400    FOUCCE: MVI C,A       ;GET A POSSIBLE COMMA COUNT
3199      53420    LDA TEMP3          ;GET THE FORMAT SPECS
3200      53440    ANI 100            ;LOOK AT THE COMMA BIT
3201      53460    RNZ               ;WE ARE USING COMMAS, JUST RETURN
3202      53480    MOV C,A           ;WE AREN'T, ZERO THE COMMA COUNT
3203      53500    RET               ;ALL DONE
3204
3205      53560    ;HERE TO PRINT A NUMBER IN FIXED FORMAT
3206      53580    FOUTFX: CPI 4          ;CHECK WHAT KIND OF VALUE WE HAVE
3207      53590    MOV A,D           ;GET THE FORMAT SPECS
3208      53600    JNC FUFVX          ;WE HAVE A SNG OR A DBL
3209
3210      53620    ;HERE TO PRINT AN INTEGER IN FIXED FORMAT
3211      53640    RAR               ;CHECK IF WE HAVE TO PRINT IT IN FLOATING
3212      53660    JC FFXFL           ;POINT NOTATION
3213      53680    ;HERE TO PRINT AN INTEGER IN FIXED FORMAT-FIXED POINT NOTATION
3214      53700    LXI B,6440083+$CODE ;SET THE DECIMAL POINT COUNT TO 6 AND
3215      53720    ;COMMA COUNT TO 3
3216      53740    CALL FUFCC          ;CHECK IF WE DON'T HAVE TO USE THE COMMAS
3217      53760    POP D             ;GET THE FIELD LENGTHS
3218      53780    MOV A,D           ;SET IF WE HAVE TO PRINT EXTRA SPACES BECAUSE
3219      53800    SUI 5             ;THE FIELD IS TOO BIG
3220      53820    CP FUTZER         ;WE DO, PUT IN ZEROS, THEY WILL LATER BE
3221      53840    PUSH D             ;MOVED TO THE SPACES OR ASTERISKS BY FOUTZS
3222      53860    CALL FUTC1          ;CONVERT THE NUMBER TO DECIMAL DIGITS
3223      53880    POP D             ;GET THE FIELD LENGTHS BACK
3224      53920    ORA E             ;DO WE NEED A DECIMAL POINT? HERE #0#
3225      53940    CZ FUFXIS          ;WE DON'T, BACKSPACE OVER IT, AT FUFXIS,
3226      53960    ;WE DON'T CARE ABOUT B, WE ONLY WANT THE
3227      53980    ;"DCX H", LATER ON WE WILL DO A "MVI M,0"
3228      54000    ;AND COVER UP WHAT WAS IN B.
3229      54020    CNZ FUTZER         ;FUFXIS PRESERVES THE CONDITION CODES

```

```

3230      54040    ;IF WE DO HAVE DECIMAL PLACES, FILL THEM UP
3231      54060    ;WITH ZEROS
3232      54080    ;FALL IN AND FINISH UP THE NUMBER
3233
3234      54120    ;HERE TO FINISH UP A FIXED FORMAT NUMBER
3235      54140    FOUTTS1 PUSH H          ;SAVE BUFFER POINTER
3236      54160    CALL FOUTZS         ;ZERO SUPPRESS THE NUMBER
3237      54180    POP H             ;GET THE BUFFER POINTER BACK
3238      54200    JZ FFXIX1          ;CHECK IF WE HAVE A.TRAILING SIGN
3239      54220    MOV M,R             ;WE DO, PUT THE SIGN IN THE BUFFER
3240      54240    INX H             ;INCREMENT THE BUFFER POINTER
3241      54260    FFXIX1: MVI M,0       ;PUT A ZERO AT THE END OF THE NUMBER
3242
3243      54300    ;HERE TO CHECK IF A FIXED FORMAT-FIXED POINT NUMBER OVERFLOWED ITS
3244      54320    #FIELD LENGTH
3245      54340    #ID = THE B IN THE FORMAT SPECIFICATION
3246      54360    #THIS ASSUMES THE LOCATION OF THE DECIMAL POINT IS IN TEMP2
3247      54380    LXI H,FBUFFR         ;GET THE POINTER TO THE BEGINNING
3248      54400    FOUBE1: INX H          ;INCREMENT POINTER TO THE NEXT CHARACTER
3249      54420    FOUBE1: LDA TEMP2         ;GET THE LOCATION OF THE DECIMAL POINT
3250      54440    SUB L             ;FIGURE OUT HOW MUCH SPACE WE ARE TAKING
3251      54460    SUB D             ;IS THIS THE RIGHT AMOUNT OF SPACE TO TAKE?
3252      54480    MOV A,M           ;YES, WE ARE DONE, RETURN FROM FOUT
3253      54500    ;NO, WE MUST HAVE TOO MUCH SINCE WE STARTED
3254      54520    ;CHECKING FROM THE BEGINNING OF THE BUFFER
3255      54540    ;THE BUFFER FIELD MUST BE SMALL ENOUGH TO FIT IN
3256      54560    ;THE BUFFER, GET THE NEXT CHARACTER IN
3257      54580    ;THE BUFFER
3258      54600    CPI "# "          ;IF IT IS A SPACE OR AN ASTERISK, WE CAN
3259      54620    JZ FOUBE1         ;IGNORE IT AND MAKE THE FIELD SHORTER WITH
3260      54640    CPI "*"           ;NO NLL EFFECTS
3261      54660    JZ FOUBE1
3262      54680    DCX H             ;MOVE THE POINTER BACK ONE TO READ THE
3263      54700    ;CHARACTER WITH CMGET
3264      54720    PUSH H             ;SAVE THE POINTER
3265
3266      54760    ;HERE WE SEE IF WE CAN IGNORE THE LEADING ZERO BEFORE A DECIMAL POINT.
3267      54780    ;THIS OCCURS IF WE SEE THE FOLLOWING: (IN ORDER)
3268      54800    ;A SIGN (EITHER "+" OR "-") [OPTIONAL]
3269      54820    ;A DOLLAR SIGN [OPTIONAL]
3270      54840    ;A DECIMAL POINT [MANDATORY]
3271      54860    ;ANOTHER DIGIT [MANDATORY]
3272      54880    ;IF YOU SEE A LEADING ZERO, IT MUST BE THE ONE BEFORE A DECIMAL POINT
3273      54900    ;FOR ELSE FOUTZS WOULD HAVE SUPPRESSED IT, SO WE CAN JUST "INX H"
3274      54940    ;OVER THE CHARACTER FOLLOWING THE ZERO, AND NOT CHECK FOR THE
3275      54960    FOUBE2: PUSH PSW        ;INCREMENT POINT EXPLICITLY,
3276      54980    ;PUT THE LAST CHARACTER ON THE STACK, THE
3277      55000    ;ZERO FLAG IS SET, THE FIRST TIME THE ZERO
3278      55020    ;ZERO FLAG IS NOT SET,
3279      55040    LXI B,FOUBE2         ;GET THE ADDRESS WE GO TO IF WE SEE A CHARACTER
3280      55060    PUSH B             ;WE ARE LOOKING FOR

```

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 13-1
F4 MAC 23-AUG-64 06:08 FLOATING POINT OUTPUT ROUTINE

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 13-12
F4 MAC 23-AUG-64 06:08 FLOATING POINT OUTPUT ROUTINE

```

3335      5b140 STA TEMP3
3336      5b160 POP H
3337      5b180 UCX
3338      5b200
3339      5b220 CALL FUUT1
3340      5b240
3341      5b260 UCX H
3342      5b280 HVI H,"x"
3343      5b300 RET
3344      5b320 JHERE TO PRINT A SNG IN FIXED FORMAT--FIXED POINT NOTATION
3345      5b340 FFXFSX1: MOVKI 226,016,033,312 /GET 1E16, CHECK IF THE NUMBER IS TOO BIG
3346      5b360 CALL FCOPN
3347      5b380 JP FFXSD0
3348      5b400 LXI D,64400+2*SCODE
3349      5b420
3350      5b440
3351      5b460 JHERE TO ACTUALLY PRINT A SNG OR DBL IN E NOTATION
3352      5b480 FFXSDC: FSIGN
3353      5b500 CNZ FOUTNV
3354      5b520
3355      5b540 POP H
3356      5b560 POP B
3357      5b580 JM FXXV5
3358
3359      5b600 JHERE TO PRINT A NUMBER WITH NO FRACTIONAL DIGITS
3360      5b620 PUSH B
3361      5b640 MOV C,A
3362      5b660 MOV A,B
3363      5b680 SUB D
3364      5b700 SUB C
3365      5b720 CP FUTZER
3366      5b740 MOV A,C
3367      5b760 ADD E
3368      5b780 MOV E,C
3369      5b800 FFXXV1: SUI 3
3370      5b820 JNC FFXXV1
3371      5b840 ADI 5
3372      5b860 CALL FOUTC
3373      5b880 MOV A,E
3374      5b900 ADD D
3375      5b920 INR A
3376      5b940 MOV B,A
3377      5b960 PUSH D
3378      5b980 CALL FOUTCV
3379      5b990 POP D
3380      5b9d0 ORA E
3381      5b9f0
3382      5b9e0 CNZ FUTZRC
3383      5b9f0 PUP D
3384      5c100 ORA E
3385      5c120 CNZ FOUTDP
3386      5c140 DCR A
3387      5c160 CP FUTZER

```

```

3388      S1160    JMP     FOUTTS   JGO CHECK THE SIZE, ZERO SUPPRESS, ETC. AND
3389      S1200    SUB     D         ; FINISH THE NUMBER
3390      S1220    JHERE TO PRINT A SNG OR DBL THAT HAS FRACTIONAL DIGITS
3391      S1240    FFXXV1  MOV     E,A   ;SAVE THE EXPONENT, WE DON'T NEED WHAT IS IN E
3392      S1260    MOV     A,C   ;DIVIDE BY TEN THE RIGHT NUMBER OF TIMES SO
3393      S1280    ORA     A       ; THE RESULT WILL BE ROUNDED CORRECTLY AND
3394      S1300    CNZ     DCRART  ; HAVE THE CORRECT NUMBER OF SIGNIFICANT
3395      S1320    ADD     D       ; DIGITS
3396      S1340    PUSH    PSH   ;SAVE THIS NUMBER FOR LATER
3397      S1360    FFXXV2  CM      FINOIV  ;THIS IS THE DIVIDE LOOP
3398      S1380    JM      FFXXV2
3399      S1400    MOV     A,E   ;WE HAVE TWO CASES DEPENDING ON WHETHER THE
3400      S1420    ADD     D       ; THE NUMBER IS ,LT, ,1 OR NOT
3401      S1440    MOV     A,B
3402      S1460    JNE     1
3403      S1480    JHERE TO PRINT NUMBERS ,GE, +1
3404      S1500    SUB     E       ;PRINT SOME LEADING ZEROS IF THE FIELD IS
3405      S1520    SUB     E       ; BIGGER THAN THE NUMBER OF DIGITS WE WILL
3406      S1540    CP      FUTZER  ; PRINT
3407      S1560    POP     PSH   ;WE DON'T NEED THE NUMBER WE SAVED BEFORE
3408      S1580    MOV     B,E   ;GET ALL THE PERTINENT INFO IN B,C
3409      S1600    PUSH    B       ;SAVE THE EXPONENT AND "C" IN FIELD SPEC
3410      S1620    MOV     A,D   ;SET UP THE DECIMAL POINT COUNT
3411      S1640    ADD     D
3412      S1660    INR     A
3413      S1680    MOV     B,A
3414      S1700    MOV     A,D   ;SET UP THE COMMA COUNT
3415      S1720    ANI     2   ;THESE 2 INSTRUCTIONS MAP 6 TO 4
3416      S1740    ADI     2   ;1 AND 16 TO 2
3417      S1760    ADD     D
3418      S1780    CALL    FUOTCC  ;CHECK IF WE HAVE TO DO THE COMMA THING
3419      S1800    JNE     1   ;CONVERT THE DIGITS AND DO THE TRIMMING UP
3420      S1820    JHERE TO PRINT A NUMBER ,LT, 1
3421      S1840    FFXXV3  CALL    FUTZER  ;PUT ALL ZEROS BEFORE THE DECIMAL POINT
3422      S1860    MOV     A,C   ;SAVE C
3423      S1880    CALL    FUOTOP  ;PUT IN A DECIMAL POINT
3424      S1900    MOV     C,A   ;RESTORE C
3425      S1920    POP     P
3426      S1940    JHERE TO PRINT ,LT, 1
3427      S1960    XHA     A       ;GET THE NUMBER WE SAVED
3428      S1980    SUB     E       ;DECIMAL POINT AND THE FIRST DIGIT WE WILL
3429      S2000    SUB     D       ;PRINT. HERE THE FIELD IS BIG ENOUGH TO
3430      S2020    JMP     FFXXV5  ;HOLD ALL THE DIGITS
3431      S2040    FFXXV4  MOV     A,C   ;GO PRINT THEM
3432      S2060    SUB     D
3433      S2080    DCR     A       ;IS SMALLER THAN ALL SIGNIFICANT DIGITS IN
3434      S2100    S1800    CALL    FUTZER  ;THE NUMBER
3435      S2120    MOV     B,E   ;PRINT THE ZEROS
3436      S2140    PUSH    B       ;SAVE EXPONENT AND THE "C" IN THE FIELD SPEC
3437      S2160    MOV     B,A   ;ZERO THE DECIMAL PLACE COUNT
3438      S2180    MOV     C,A   ;ZERO THE COMMA COUNT
3439      S2200    FFXXV6  CALL    FUOTCV  ;CONVERT THE NUMBER TO DECIMAL DIGITS
3440      S2220    POP     D       ;GET THE EXPONENT AND FIELD SPEC BACK

```

```

3441      S2240    DRA    E       ;CHECK IF WE HAVE TO PRINT ANY ZEROS AFTER
3442      S2260    JZ      FFXXV7  ;THE LAST DIGIT
3443      S2280    ADD     D       ;CHECK IF THERE WERE ANY DECIMAL PLACES AT ALL
3444      S2300    DCR     A       ;PRINT SOME MORE TRAILING ZEROS
3445      S2320    OCK    A
3446      S2340    CP      FUTZER
3447      S2360    JMP     FOUTTS  ;FINISH UP THE NUMBER
3448      S2380    JHERE WERE NO DECIMAL PLACES, IGNORE ALL DIGITS AFTER THE DECIMAL
3449      S2400    JPOINT
3450      S2420    FFXXV7  LHMD    TEMP2  ;THE END OF THE NUMBER IS WHERE THE DP IS
3451      S2440    JNE     1   ;FINISH UP THE NUMBER
3452
3453      S2480    JHERE TO PRINT AN INTEGER IN FIXED FORMAT--FLOATING POINT NOTATION
3454      S2500    FFXFL1  PUSH    M       ;SAVE THE BUFFER POINTER
3455      S2520    PUSH    D       ;SAVE THE FORMAT SPECS
3456      S2540    CALL    CONSI   ;CONVERT THE INTEGER TO A SNG
3457      S2560    POP     D       ;GET THE FORMAT SPECS BACK
3458      S2580    POP     M       ;GET THE BUFFER POINTER BACK
3459      S2600    XRA     A       ;SET FLAG TO PRINT THE NUMBER AS A SNG
3460      S2620    S1800    JHERE TO PRINT A SNG OR DBL IN FIXED FORMAT--FLOATATING POINT NOTATION
3461      S2640    S1860    FFXXFL1 JZ      FFXXFL  ;JIF WE HAVE A SNG, SET THE RIGHT FLAGS
3462      S2660    MOV     E,20  ;WE HAVE A DBL, GET HOW MANY DIGITS WE HAVE
3463      S2680    XWD    1000,001 ;JWXN  BY GETTER THE NEXT TWO BYTES
3464      S2700    FFXXFL1 JI      E,6   ;JIF WE HAVE A DBL, GET HOW MANY DIGITS WE PRINT
3465      S2720    PSIGN
3466      S2740    CNZ     FOUTNV  ;SEE IF WE HAVE ZERO
3467      S2760    DCR     A       ;IF NOT, NORMALIZE THE NUMBER SO ALL DIGITS TO
3468      S2780    CNZ     FOUTNV  ;BE PRINTED ARE IN THE INTEGER PART
3469      S2800    POP     M       ;GET THE BUFFER POINTER BACK
3470      S2820    POP     B       ;GET THE FIELD LENGTH SPECS
3471      S2840    PUSH    PSH   ;SAVE THE EXPONENT
3472      S2860    MOV     A,C   ;CALCULATE HOW MANY SIGNIFICANT DIGITS WE MUST
3473      S2880    DCR     A       ;PRINT
3474      S2900    PUSH    PSH   ;SAVE THE "C" FIELD SPEC FOR LATER
3475      S2920    CNZ     DCRART
3476      S2940    ADD     B
3477      S2960    MOV     C,A
3478      S2980    MOV     A,D   ;GET THE FIELD SPEC
3479      S3000    ANI     4   ;SEE IF THE SIGN IS A TRAILING SIGN
3480      S3020    CPI     1   ;SET CARRY IF A IS ZERO
3481      S3040    SBS    D,A   ;SET D=0 IF WE HAVE A TRAILING SIGN,
3482      S3060    POP     D,A   ;D=577 IF WE DO NOT
3483      S3080    JHERE TO PRINT A SNG OR DBL IN FIXED FORMAT--FLOATATING POINT NOTATION
3484      S3100    S1800    FFXXFL1 CM      FINOIV  ;JIF WE HAVE LESS THAN E, THEN WE MUST GET RID
3485      S3120    S1860    JFXXFL1 ;OF SOME BY DIVIDING BY TEN AND ROUNDING
3486      S3140    PUSH    B       ;SAVE THE "B" FIELD SPEC & # OF SIG DIGITS
3487      S3160    MOV     A,B   ;SET THE DECIMAL PLACE COUNT
3488      S3180    INR     A
3489      S3200    SUB     D
3490      S3220    MOV     B,A   ;TAKE INTO ACCOUNT IF THE SIGN IS TRAILING
3491
3492      S3240    S1800    FFXXFL1 CM      FINOIV  ;JIF WE HAVE LESS THAN E, THEN WE MUST GET RID
3493

```

```

3494      59300  MVI   C,0          ;SET COMMA COUNT TO ZERO, THE COMMA SPEC IS
3495      59320  PUSH  D          ; IGNORED, SAVE TRAILING SIGN INFO
3496      59340  CALL   FOUTCV     ;CONVERT THE NUMBER TO DECIMAL DIGITS
3497      59360  POP   D          ;GET THE TRAILING SIGN INFO BACK
3498      59380  POP   B          ;GET A OF SIG DIGITS AND "B" FIELD SPEC BACK
3499      59400  MOV   A,C         ;PRINT TRAILING ZEROS IF THE FIELD LENGTH IS
3500      59420  SUB   E          ; LONGER THAN THE NUMBER OF DIGITS WE CAN PRINT
3501
3502      59440  CP    FUTZRC     ;THE DECIMAL POINT COULD COME OUT IN HERE
3503      59460  POP   PSW         ;GET THE "C" FIELD SPEC BACK
3504      59480  CZ    FUFIX1S     ;IF C#0, THE LAST THING WAS A DECIMAL POINT,
3505      59500  ADD   E          ;SO IGNORE IT, ALL WE CARE ABOUT IS THE
3506      59520  SUB   B          ; "UCK H AND NOT THE "MOV M,B" AT FOFIX1S
3507      59540  POP   PSW         ;GET THE EXPONENT BACK
3508      59560  ADD   E          ;SCALE IT CORRECTLY
3509      59580  SUB   B          ;
3510      59600  SUB   D          ;
3511      59620  PUSH  B          ;SAVE THE "B" FIELD SPEC
3512      59640  CALL   FUFLON     ;PUT THE EXPONENT IN THE BUFFER
3513      59660  XCHG  (HL)        ;GET THE POINTER TO THE END IN (HL)
3514      59680  POP   D          ;IN CASE WE HAVE A TRAILING SIGN
3515      59700  POP   D          ;GET THE "B" FIELD SPEC IN D, PUT ON A POSSIBLE
3516
3517      59720  JMP   FOUTTS     ;TRAILING SIGN AND WE ARE DONE
3518      59740
3519      59760  INORMALIZE THE NUMBER IN THE FAC SO ALL THE DIGITS ARE IN THE INTEGER
3520  FPART, RETURN THE BASE 10 EXPONENT IN A
3521  DUE ARE LEFT UNALTERED
3522      59780  FOUTNVI PUSH  D          ;SAVE (DE)
3523      59790  CPI   6          ;GET WHAT KIND OF VALUE WE HAVE
3524      59860  JNZ   FOUTNDO     ;WE HAVE A DBL
3525      59900  INORMALIZE A SNG
3526      59920  XRA   A          ;ZERO THE EXPONENT
3527      59940  PUSH  PSW         ;SAVE IT
3528      59960  CALL   FOUNSC     ;IS THE FAC TOO BIG OR TOO SMALL?
3529      59980  FOUNS11 MOVRI 221,103,117,370 ;GET 99999,9499 TO SEE IF THE FAC IS BIG
3530      60000  CALL   FCMP        ;ENOUGH YET
3531      60020  JPD   FOUNS3     ;IT IS, WE ARE DONE
3532      60040  POP   PSW         ;IT ISN'T, MULTIPLY BY TEN
3533      60060  CALL   FINHLT     ;SAVE THE EXPONENT AGAIN
3534      60080  PUSH  PSW         ;NOW SEE IF IT IS BIG ENOUGH
3535  FOUNS12 JMP   FOUNS1      ;THE FAC IS TOO BIG, GET THE EXPONENT
3536      60120  POP   PSW         ;DIVIDE IT BY TEN
3537      60140  CALL   FINDIV     ;SAVE THE EXPONENT AGAIN
3538      60160  PUSH  PSW         ;ISER IF THE FAC IS SMALL ENOUGH
3539      60180  CALL   FOUNSC     ;WE ARE DONE, GET THE EXPONENT BACK
3540      60200  FOUNS31 POP   PSW         ;POP D
3541      60220  POP   D          ;GET (DE) BACK
3542      60240  RET             ;ALL DONE
3543      60260  INHERE TO SEE IF THE FAC IS SMALL ENOUGH YET
3544      60280  FOUNS11 MOVRI 224,164,043,367 ;GET 999999,499 TO SEE IF THE FAC IS TOO BIG
3545      60300  CALL   FCMP        ;HERE TO NORMALIZE A DBL NUMBER
3546

```

```

3547      60320  PUP   H          ;GET THE RETURN ADDRESS OFF THE STACK
3548      60340  JPO   FOUNS2     ;IT IS TOO BIG, MAKE IT SMALLER
3549      60360  PCNL  FOUNS2     ;IT IS SMALL ENOUGH, RETURN
3550      60380
3551      60400  >                ;HERE TO NORMALIZE A DBL NUMBER
3552      60420  PAGE

```

```

3553          60440 SUBTL EXPONENTIATION AND THE SQUARE ROOT FUNCTION
3554          60460 IFE EXTFNC,<
3555          60480 JSQR SQR ROUTINE --- X=SQR(A)
3556          60500 ;WE FIRST SCALE THE ARGUMENT TO BETWEEN .5 AND 2 BY LOOKING AT THE
3557          60520 ;EXPONENT AND USING SQR(M*2^(Z*N))=2^N*SQR(M), THEN NEWTON'S METHOD
3558          60540 ;IS USED TO COMPUTE SQR(M), THE EXPONENT IS SAVED TO SCALE THE
3559          60560 ;RESULT AT THE END.
3560          60580 ;NEWTON'S METHOD FOR SQUARE ROOT:
3561          60600 ; 1) X(0)=A
3562          60620 ; 2) X(N+1)=(X(N)*A/X(N))/2
3563          60640 SQR: FSIGN ;CHECK FOR ERROR CONDITION
3564          60660 JM FCERR ;CAN'T TAKE SQR OF NEGATIVE NUMBER
3565          60680 RZ ;SET EXPONENT TO 0
3566          60700 LXI H,FAC ;SCALE ARGUMENT TO BETWEEN .5 AND 2
3567          60720 MOV A,M ;GET EXPONENT
3568          60740 RAH ;SET EXPONENT OF SCALE FACTOR
3569          60760 PUSH PSH ;USE SQR(M*2^(Z*N))=2^N*SQR(M)
3570          60780 PUSH M ;SAVE IT
3571          60800 MVI A,100 ;SAVE POINTER TO EXPONENT
3572          60820 MVI A,100 ;SET EXPONENT TO SCALED DOWN NUMBER
3573          60840 RAL ;GET EXPONENT
3574          60860 MVI H,A ;REPLACE IT
3575          60880 LXI H,FBUFFR ;SAVE A
3576          60900 CALL MOVHF
3577          60920 MVI A,4 ;SET ITERATION COUNT
3578          60940 SQR1: PUSH PSW ;SAVE COUNT
3579          60960 CALL PUSHF ;SAVE X(N)
3580          60980 LXI H,FBUFFR ;COMPUTE A/X(N)
3581          61000 CALL MOVFR ;GET A IN THE REGISTERS
3582          61020 CALL FDIV
3583          61040 POP PSW ;ADD IN X(N)
3584          61060 CALL FAD0 ;DIVIDE BY 2
3585          61080 LXI H,PHALF ;GET COUNT
3586          61100 CALL FMULTS ;GET EXPONENT OF ANSWER
3587          61120 POP PSW ;ARE WE DONE?
3588          61140 DCR A ;NO, DO MORE ITERATIONS
3589          61160 JNZ SQR1 ;YES, SET EXPONENT OF ANSWER
3590          61180 POP PSW ;GET SCALE FACTOR
3591          61200 ADD S00 ;CONVERT TO AN EXPONENT
3592          61220 ADD S00 ;ADD EXPONENT IN
3593          61240 ADD M ;REPLACE EXPONENT
3594          61260 MOV M,A
3595          61280 RET> ;FALL DONE
3596
3597
3598          61380 IFN EXTFNC,<
3599          61360 ;SUBROUTINE FOR FPWR, ATN
3600 002340# 001000 000041 61380 PSHNEG1 LXI M,NEG ;GET THE ADDRESS OF NEG
3601 002341# 000000 001175# 61400 XTHL ;SWITCH RET ADDR AND ADDR OF NEG
3602 002342# 000000 002307# 61420 PCML ;RETURN, THE ADDRESS OF NEG IS ON THE STACK
3603 002343# 001000 000343 61400
3604 002344# 001000 000351 61420
3605

```

```

3606          61480 JSQR SQR FUNCTION
3607          61500 ;WE USE SQR(X)=X^.5
3608          61520 SQR: CALL PUSHF ;SAVE ARG
3609 002345# 001000 000315 61540 LXI H,PHALF ;GET 1/2
3610 002346# 000000 001209# 61560 CALL MOVFM ;SQR(X)=X^.5
3611 002347# 000000 002341# 61580 FPWR1: PURR ;GET ARG IN REGISTERS, ENTRY TO FPWR IF
3612 002350# 001000 000041 61600 ;ARGUMENT IS ON STACK, FALL INTO FPWR
3613 002351# 000000 002312# 61620
3614 002352# 000000 002346# 61640
3615 002353# 000000 002322# 61660
3616 002354# 000000 002355# 61680 ;EXponentiation --- XY
3617 002355# 000000 002351# 61700 ;IF X=0, IF SO, THE RESULT IS 1.
3618 002356# 001000 000301 61720 ;INEXT, WE CHECK IF X#0, IF SO, THE RESULT IS 0.
3619 002357# 000000 000321 61740 ;THEN WE CHECK IF X IS POSITIVE, IF NOT, WE CHECK THAT Y IS A
3620          61760 ;NEGATIVE INTEGER, AND WHETHER IT IS EVEN OR ODD. IF Y IS A NEGATIVE
3621          61780 ;INTEGER, WE NEGATE X, LOG WILL GIVE AN FC ERROR WHEN WE CALL
3622          61800 ;FPWR, IF X IS NEGATIVE AND IS ODD, WE PUSH THE ADDRESS OF NEG ON THE
3623          61820 ;STACK, SO WE WILL RETURN TO IT AND GET A NEGATIVE RESULT, TO COMPUTE
3624          61840 ;THE RESULT WE USE XY=EXP(Y*LOG(X))
3625 002360# 001000 000357 61860 FPWR1: FSIGN ;SEE IF Y IS ZERO
3626 002361# 001000 000312 61880 JZ EXP ;IT IS, RESULT IS ONE
3627 002362# 000000 002456# 61900 MOV A,B ;ISEE IF X IS ZERO
3628 002363# 000000 002354# 61920 ORA A ;TURN THE ZERO FLAG OFF
3629 002365# 001000 000176# 61940 JZ ZERO0 ;IT IS, RESULT IS ZERO
3630 002366# 001000 000312 62020 CALL MOVRF ;GET Y IN THE REGISTERS
3631 002367# 000000 000174# 62040 JP FPWR1 ;NO PROBLEMS IF X IS POSITIVE
3632 002370# 000000 002356# 62060
3633 002360# 001000 000357 62080 PUSHR ;SAVE X ON STACK
3634 002361# 001000 000312 62100 MOV A,C ;CHECK THE SIGN OF X
3635 002362# 000000 002456# 62000 ORI 177 ;TURN THE ZERO FLAG OFF
3636 002363# 000000 002354# 62020
3637 002371# 001000 000171 62040 CALL MOVRF ;GET Y BACK
3638 002374# 001000 000356# 62060
3639 002375# 001000 000177 62080
3640 002376# 001000 000315 62100
3641 002370# 000000 002356# 62120
3642 002371# 001000 000325 62140
3643 002372# 001000 000305 62160
3644 002373# 001000 000171 62180
3645 002374# 001000 000356# 62200
3646 002375# 001000 000177 62220
3647 002376# 001000 000315 62240
3648 002377# 000000 001240# 62260
3649 002400# 000000 002357# 62280
3650 002401# 001000 000356# 62300
3651 002402# 000000 002422# 62320
3652 002403# 000000 002377# 62340
3653 002404# 001000 000325 62360
3654 002405# 001000 000305 62380
3655 002406# 001000 000315 62400
3656 002407# 000000 001445# 62420
3657 002416# 000000 002402# 62440
3658 002411# 001000 000301 62460

```

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 14-2
F4 MAC 23-AUG-64 06:08 EXPONENTIATION AND THE SQUARE ROOT FUNCTION

3659	002412*	001000	000321					
3660	002413*	001000	000365	62120	PUSH	PSW	I SAVE LO OF INT FOR EVEN AND ODD INFORMATION	
3661	002414*	001000	000315	62140	CALL	FCOMP	I SEE IF HE HAVE AN INTEGER	
3662	002415*	000000	001317*					
3663	002416*	000000	002407/*					
3664	002417*	001000	000341	62160	POP	H	I GET EVEN-ODD INFORMATION	
3665	002420*	001000	000174	62180	MVR	A,H	I PUT EVEN-ODD FLAG IN CARRY	
3666	002421*	001000	000000	62200	RAR			
3667	002424*	001000	000441	62220	FPHR11	POP	I GET X BACK IN FAC	
3668	002423*	001000	000442	62240	SHLD	FAC=1	I STORE HOTS	
3669	002424*	777777	777777					
3670	002425*	000000	002415*					
3671	002426*	001000	000341	62260	PUP	H	I GET LO'S OFF STACK	
3672	002427*	001000	000042	62280	SHLD	FACLO	I STORE THEM IN FAC	
3673	002428*	000000	000000					
3674	002431*	001000	000244*					
3675	002434*	001000	000334	62300	CC	PSHNEG	I NEGATE NUMBER AT END IF Y WAS ODD	
3676	002435*	000000	002340*					
3677	002434*	000000	002436*					
3678	002435*	001000	000314	62320	CZ	NEG	I NEGATE THE NEGATIVE NUMBER	
3679	002436*	000000	001175*					
3680	002437*	000000	002435*					
3681	002438*	001000	000000	62340	FPHR21	PUSHR	I SAVE Y AGAIN	
3682	002441*	001000	000305					
3683	002442*	001000	000515	62360	CALL	LOG	I COMPUTE EXP(Y*LOG(X))	
3684	002443*	000000	000421*					
3685	002444*	000000	002436*					
3686	002445*	001000	000301	62380	POPN		I IF X WAS NEGATIVE AND Y NOT AN INTEGER THEN	
3687	002446*	001000	000321					
3688	002447*	001000	000315	62400	CALL	FMULT>	I LOG WILL BLOW HIM OUT OF THE WATER	
3689	002450*	000000	000517/*					
3690	002451*	000000	002443*					
3691				62420	J	JMP	EXP	
3692				62440	PAGE			

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 15
F4 MAC 23-AUG-64 06:08 EXPONENTIAL FUNCTN .

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 15-1
F4 MAC 23-AUG-64 06108 EXPONENTIAL FUNCTION

3746 002524* 001000 000321
3747 002525* 001000 000369
3748 002526* 001000 000315
3749 002527* 000000 00015
3750 002530* 000000 0002526*
3751 002531* 000000 00015
3752 002532* 000000 001175*
3753 002533* 000000 002527*
3754 002534* 001000 000041
3755 002535* 000000 002525*
3756 002536* 000000 00253c*
3757 002537* 001000 000515
3758 002540* 000000 002632*
3759 002541* 000000 000035*
3760 002542* 001000 000021
3761 002543* 000000 001477*
3762 002544* 000000 002540*
3763 002545* 001000 000301
3764 002546* 001000 000114
3765 002547* 001000 000303
3766 002550* 000000 000517*
3767 002551* 000000 000543*
3768
3769
3770 002552* 000000 000010
3771 002553* 000000 000100
3772 002554* 000000 000056
3773 002555* 000000 000224
3774 002556* 000000 000164
3775 002557* 000000 000164
3776 002558* 000000 000117
3777 002559* 000000 000056
3778 002560* 000000 000167
3779 002561* 000000 000156
3780 002564* 000000 000002
3781 002565* 000000 000210
3782 002566* 000000 000074
3783 002567* 000000 000346
3784 002568* 000000 000240
3785 002571* 000000 000052
3786 002572* 000000 000174
3787 002573* 000000 000120
3788 002574* 000000 000252
3789 002575* 000000 000252
3790 002576* 000000 000076
3791 002577* 000000 000377
3792 002580* 000000 000377
3793 002601* 000000 002177
3794 002602* 000000 000177
3795 002603* 000000 000000
3796 002604* 000000 000000
3797 002605* 000000 000000
3798 002606* 000000 000201
630000 PUSH PSH INPUT SCALE FACTOR BACK ON STACK
63020 CALL FSUB FSUBTRACT ORIGINAL ARG
63040 CALL NEG
63060 LXI H,EXPCON EVALUATE THE APPROXIMATION POLYNOMIAL
63080 CALL PULY
63100 LXI D,SCODE /MULTIPLY BY 2 * (B=1) INSTEAD OF JUST
63120 POP B ; ADDING IT TO THE EXPONENT SO FMULT
63140 MOV C,D ; WILL CHECK FOR EXPONENT OVERFLOW
63160 JMP FMULT
63200 EXPCON //CONSTANTS FOR EXP
63220 EXPCON 10 DEGREE
63240 100 ; .0001413161
63260 056
63280 224
63300 164
63320 100 ; .001329882
63340 117
63360 056
63380 167
63400 156 ; .00830136
63420 002
63440 210
63460 172
63480 346 ; .04165735
63500 240
63520 052
63540 174
63560 120 ; .1666653
63580 252
63600 252
63620 176
63640 176
63660 377 ; .4999999
63680 177
63700 177
63720 000 ; =1.0
63740 000
63760 200
63780 201
63800 PAGE

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 15-2
F4 MAC 23-AUG-64 06108 EXPONENTIAL FUNCTION

3799 002607* 000000 000000 000 1.
3800 002610* 000000 000000 000
3801 002611* 000000 000000 000
3802 002612* 000000 000201 000 201
3803 002608 PAGE

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 16

POLYNOMIAL EVALUATOR AND THE RANDOM NUMBER GENERATOR

```

3804          63900  SUBTTL POLYNOMIAL EVALUATOR AND THE RANDOM NUMBER GENERATOR
3805          63920  IFN      EXTFCNC,<
3806          63940  JEVALUATE P(X^2)*X
3807          63960  JPOINTER TO DEGREE+1 IS IN (HL)
3808          63980  JTHE CONSTANTS FOLLOW THE DEGREE
3809          64000  JCONSTANTS SHOULD BE STORED IN REVERSE ORDER, FAC HAS X
3810          64020  JWE COMPUTE C0+C1*X+C2*X^2+C3*X^3+...+C(N)*X^(2*N+1)
3811          64040  J C0*X+C1*X^3+C2*X^5+C3*X^7+...+C(N)*X^(2*N+1)
3812          64060  POLYXI CALL    PUSHF   JSAVE X
3813  002615*  001000  000315
3814  002615*  000000  001205*
3815  002615*  000000  002550*
3816  002615*  001000  000201
3817  002615*  000000  000151*
3818  002621*  001000  000525
3819  002622*  001000  000345
3820  002623*  001000  000315
3821  002624*  000000  001240*
3822  002625*  000000  002617*
3823  002625*  001000  000315
3824  002627*  000000  000517*
3825  002630*  000000  002624*
3826  002631*  001000  000341
3827          64180  POP     H      JGET CONSTANT POINTER
3828          64200  CALL    FMULT   JFALL INTO POLY
3829
3830          64260  JPOLYNOMIAL EVALUATOR
3831          64280  JPOINTER TO DEGREE+1 IS IN (HL), IT IS UPDATED
3832          64300  JTHE CONSTANTS FOLLOW THE DEGREE
3833          64320  JCONSTANTS SHOULD BE STORED IN REVERSE ORDER, FAC HAS X
3834          64340  JWE COMPUTE
3835          64360  J C0*C1*X+C2*X^2+C3*X^3+...+C(N-1)*X^(N-1)+C(N)*X^N
3836  002632*  001000  000315
3837  002633*  000000  001205*
3838  002634*  000000  002627*
3839  002635*  001000  000176
3840  002636*  001000  000143
3841  002637*  001000  000315
3842  002640*  000000  001242*
3843  002641*  000000  002633*
3844  002642*  001000  000006
3845  002643*  001000  000361
3846  002644*  001000  000301
3847  002645*  001000  000321
3848  002646*  001000  000375
3849  002647*  001000  000310
3850  002650*  001000  000325
3851  002651*  001000  000303
3852  002652*  001000  000365
3853  002653*  001000  000345
3854  002654*  001000  000315
3855  002655*  000000  000517*
3856  002656*  000000  002640*

```

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 16-1

POLYNOMIAL EVALUATOR AND THE RANDOM NUMBER GENERATOR

```

3857  002657*  001000  000341
3858  002660*  001000  000315
3859  002661*  000000  001243*
3860  002662*  000000  002655*
3861          64680  PUSH   H      JSTORE LOCATION OF CONSTANTS SO FADD AND FMULT
3862  002663*  001000  000345
3863  002664*  001000  000315
3864  002665*  000000  000205*
3865  002666*  000000  002661*
3866  002667*  001000  000341
3867  002670*  001000  000303
3868  002671*  000000  002642*
3869  002672*  000000  002656*
3870
3871          64800  JPSUEDO-RANDOM NUMBER GENERATOR
3872          64820  JIF ARG=0, THE LAST RANDOM NUMBER GENERATED IS RETURNED
3873          64840  JIF ARG<LT, 0, A NEW SEQUENCE OF RANDOM NUMBERS IS STARTED
3874          64860  JJUST USE THE DOCUMENTATION
3875          64880  JTO FORM THE NEXT RANDOM NUMBER IN THE SEQUENCE, WE MULTIPLY THE
3876          64900  JPREVIOUS RANDOM NUMBER BY A RANDOM CONSTANT, AND ADD IN ANOTHER
3877          64920  JRANDOM CONSTANT, THEN THE HO AND LO BYTES ARE SWITCHED, THE
3878          64940  JEXPONENT IS PUT WHERE IT WILL BE SHIFTED IN BY NORMAL, AND THE
3879          64960  JEXPONENT IN THE FAC SET TO 200 SO THE RESULT WILL BE LESS THAN 1.
3880          64980  JTHIS IS THEN NORMALIZED AND SAVED FOR THE NEXT TIME,
3881          65000  JTHE HO AND LO BYTES ARE SWITCHED SO WE HAVE A RANDOM CHANCE OF
3882          65020  JGETTING A NUMBER LESS THAN OR GREATER THAN .5
3883  002673*  65240  RNDI
3884  002673*  65680  IFN      LENGTH=2,<
3885  002673*  001000  000357
3886  002673*  000000  000357
3887  65100  IFE      LENGTH=2,<
3888          65120  CALL    VSIGN   JGET SIGN OF ARG
3889          65140  PUSH   PSW    JSAVE THE SIGN
3890          65160  CM     PRECSNG JIF SIGN IS NEGATIVE, FORCE IT TO BE A SNG
3891          65180  MVI    A,1     JSINCE WE WILL USE IT
3892          65200  STA    VALTYP JMAKE SURE THE RESULT IS "SINGLE PRECISION"
3893          65220  POP    PSW    JGET THE SIGN BACK
3894  002674*  001000  000372
3895  002675*  000000  002725*
3896  002676*  000000  002711*
3897  002677*  001000  000841
3898  002678*  001000  000357*
3899  002679*  000000  002675*
3900  002679*  001000  000315
3901  002679*  000000  001222*
3902  002679*  000000  002709*
3903  002679*  001000  000310
3904          65300  RZ      JRETURN LAST NUMBER GENERATED IF ZERO
3905          65320  IFE      EXTFCNC,<
3906          65340  CALL    FMULTS> JMULTIPLY BY CONSTANT A
3907  002706*  001000  000315
3908  002707*  000000  001243*
3909  002710*  000000  002705*

```

MATHPK FOR BASIC MCS 8880 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 16-2
F4 MAC 23-AUG-64 06108 POLYNOMIAL EVALUATOR AND THE RANDOM NUMBER GENERATOR

3910 002711' 001000 000315 65400 CALL FMULT>
3911 002712' 001000 000517' 65420 MOVR1 150,050,261,106 !ADD IN CONSTANT OF ORDER 2^m(-24)
3912 002713' 001000 002707' 65420
3913 002714' 001000 000001 65420
3914 002715' 001000 000500 65420
3915 002715' 001000 000150 65420
3916 002717' 001000 000021 65420
3917 002720' 001000 000100 65420
3918 002721' 001000 000261 65420
3919 002722' 001000 000315 65440 CALL FADD
3920 002723' 001000 000025' 65440
3921 002724' 001000 002716' 65440
3922 002725' 001000 000315 65460 RND11 CALL MOVRF ;SWITCH MO AND LO BYTES,
3923 002725' 001000 000149' 65460
3924 002727' 001000 002723' 65460
3925 002730' 001000 000173 65480 MOV A,E ;GET LO
3926 002731' 001000 000131 65500 MOV E,C ;PUT MO IN LO BYTE
3927 002732' 001000 000117 65520 MOV C,A ;PUT LO IN MO BYTE
3928 002733' 001000 000066 65540 MVI R,200 ;MAKE RESULT POSITIVE
3929 002734' 001000 000000 65540
3930 002735' 001000 000000 65560 DCX H ;GET POINTER TO EXPONENT
3931 002736' 001000 000106 65580 MOV B,M ;PUT EXPONENT IN OVERFLOW POSITION
3932 002737' 001000 000066 65600 MVI R,200 ;SET EXP SO RESULT WILL BE BETWEEN 0 AND 1
3933 002740' 001000 000200 65600
3934 002741' 001000 000315 65620 CALL NORMAL ;NORMALIZE THE RESULT
3935 002742' 001000 000146' 65620
3936 002743' 001000 000276' 65640 LXI H,RNUX ;SAVE RANDOM NUMBER GENERATED FOR NEXT
3937 002744' 001000 000041 65640
3938 002745' 001000 000152' 65640
3939 002746' 001000 000242' 65660
3940 002747' 001000 000303 65660 JMP MOVMF ;TIME
3941 002750' 001000 001254' 65660
3942 002751' 001000 002745' 65660
3943 65700 ;CONSTANTS AND STORAGE FOR RND
3944 002752' 001000 000122 65700 RNDX1 122 ;LAST RANDOM NUMBER GENERATED, BETWEEN 0 AND 1
3945 002753' 001000 000307' 65740 122
3946 002754' 001000 000117 65760 117
3947 002755' 001000 000200 65780 200
3948 002756' 001000 000172 65800 172 ;RANDOM NUMBER OF ORDER 2^m24
3949 002757' 001000 000104 65820 104
3950 002757' 001000 000065 65840 065
3951 002758' 001000 000230 65860 230
3952 002761' 001000 000230 65880 PAGE

MATHPK FOR BASIC MCS 8880 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 17
F4 MAC 23-AUG-64 06108 SINE, COSINE AND TANGENT FUNCTIONS

3954 65900 SUBTL SINE, COSINE AND TANGENT FUNCTIONS
3955 65920 IFN EXTFNC,<
3956 65940 ;SINE FUNCTION
3957 65960 ;COSINE FUNCTION
3958 002762' 001000 000041 65980 COS1 LXI H,PI2 ;ADD PI/2 TO FAC
3959 002763' 001000 003970' 65980
3960 002764' 001000 002750' 66000 CALL FADDS>
3961 002765' 001000 000315 66000
3962 002766' 001000 000003' 66000
3963 002767' 001000 002765' 66000
3964 66020 ;FALL INTO SIN
3965
3966
3967 66080 ;SINE FUNCTION
3968 66100 ;IDEA1 USE IDENTITIES TO GET FAC IN QUADRANTS I OR IV
3969 66120 ;THE FAC IS DIVIDED BY 2*PI AND THE INTEGER PART IS IGNORED BECAUSE
3970 66140 ;JSIN(X+2*PI)=JSIN(X), THEN THE ARGUMENT CAN BE COMPARED WITH PI/2 BY
3971 66160 ;COMPARING THE RESULT OF THE DIVISION WITH PI/2/(2*PI)=1/4.
3972 66180 ;IDENTITIES ARE THEN USED TO GET THE RESULT IN QUADRANTS I OR IV,
3973 66200 ;THE APPROXIMATION POLYNOMIAL IS THEN USED TO COMPUTE SIN(X).
3974 002770' 001000 000315 66220 SIN1 PUSHF ;DIVIDE FAC BY 2*PI
3975 002771' 001000 001205' 66220
3976 002772' 001000 002766' 66240 MOVRI 203,111,017,333 ;AFTER DIVIDING BY 2*PI, RESULT IS
3977 002773' 001000 000001 66240
3978 002774' 001000 000111 66240
3979 002775' 001000 000003 66240
3980 002776' 001000 000021 66240
3981 002777' 001000 000355 66240
3982 003000' 001000 000017 66240
3983 003001' 001000 000315 66260 CALL MOVFR ;BETWEEN 0 AND 1
3984 003002' 001000 001225' 66260
3985 003003' 001000 002771' 66280 PUPR
3986 003004' 001000 000351 66280
3987 003005' 001000 001231 66300 CALL FUIV
3988 003006' 001000 000315 66300
3989 003007' 001000 000055' 66300
3990 003010' 001000 003002' 66320 CALL PUSHF ;DISREGARD INTEGER PART SINCE SIN
3991 003011' 001000 000315 66320
3992 003012' 001000 001205' 66320
3993 003013' 001000 003007' 66340 CALL INT ;IS PERIODIC WITH PERIOD 2*PI
3994 003014' 001000 000315 66340
3995 003015' 001000 001205' 66340
3996 003016' 001000 003012' 66360 PUPR
3997 003017' 001000 000301 66360
3998 003020' 001000 000321 66380 CALL FSUB
3999 003021' 001000 000313 66380
4000 003022' 001000 000017' 66400 IFE EXTFNC,<
4001 003023' 001000 003015' 66420 LXI H,177*400+SCODE ;GET 1/4
4002 66440 MOV B,C
4003 66460 MOV E,C
4004 66480 CALL FSUB

```

F4      MAC   23=AUG=64 06:08
       SINE, CUSINE AND TANGENT FUNCTIONS

4007          66500  IFN    EXTFNC,<
4008  003024* 001000  000041  66520  LXI   H,FR4    ;SEE WHAT QUADRANT WE ARE IN
4009  003025* 000000  003674*          ;SET QUADRANT I FLAG
4010  003026* 000002  003622*          ;FIRST QUADRANT, GET BACK ORIGINAL X
4011  003027* 000000  000011*          ;SET QUADRANT II FLAG
4012  003028* 000000  003625*          ;SECOND QUADRANT, GET BACK ORIGINAL X
4013  003029* 000000  003615*          ;SET QUADRANT III FLAG
4014  003030* 001000  000057  66560  FSIGN
4015  003031* 001000  000067  66580  STC
4016  003032* 001000  000062  66600  JP    SIN1    ;SET QUADRANT I FLAG
4017  003033* 000000  003644*          ;FIRST QUADRANT, GET BACK ORIGINAL X
4018  003034* 000000  003630*          ;SET QUADRANT IV FLAG
4019  003035* 000000  000015*          ;SET QUADRANT V FLAG
4020  003036* 000000  003635*          ;SET QUADRANT VI FLAG
4021  003041* 000000  003635*          ;SET QUADRANT VII FLAG
4022  003042* 001000  000057  66640  FSIGN
4023  003043* 001000  000067  66660  ORA   A    ;CLEAR CARRY
4024  003044* 001000  000065  66680  SIN11  PUSH  PSW  ;SAVE QUADRANT FLAG
4025  003045* 001000  000064  66700  CP    NEG   ;NEGATE IF IN QUADRANTS I, II OR III
4026  003046* 000000  001175*          ;SET QUADRANT VIII FLAG
4027  003047* 000000  003640*          ;SET QUADRANT IX FLAG
4028          66720  IFE    EXTFNC,<
4029  003048* 000000  003640*          ;SET QUADRANT X FLAG
4030          66740  LXI   H,177+400+SCODE ;GET 1/4
4031          66760  MOV   D,C
4032          66780  MOV   E,C
4033          66800  CALL  FADD>
4034          66820  IFN    EXTFNC,<
4035  003050* 001000  000041  66840  LXI   H,FR4    FAUD 1/4, IN QUADRANTS II, III
4036  003051* 000000  003674*          ;USE THE IDENTITY SIN(Pi-K)=SIN(X)
4037  003052* 000000  003646*          ;IN QUADRANT IV, USE THE IDENTITY:
4038          66860  CALL  FADDS> ; SIN(X-2*pi)=SIN(X)
4039  003053* 001000  000015  66900  CALL  FADDS> ; USE THE IDENTITY
4040  003054* 000000  000003*          ;IN QUADRANT V, USE THE IDENTITY:
4041  003055* 000000  003651*          ; SIN(X+2*pi)=SIN(X)
4042  003056* 001000  000057  66920  PUP   PSW   ;GET QUADRANT FLAG
4043  003057* 000000  000015*          ;NEGATE IF IN QUADRANTS III, III OR IV
4044  003064* 000000  001175*          ;SET QUADRANT VIII FLAG
4045  003061* 000000  003654*          ;SET QUADRANT IX FLAG
4046          66960  IFE    EXTFNC,< ; EVALUATE APPROXIMATION POLYNOMIAL
4047          66980  CALL  PUSHF
4048          67000  CALL  MOVRF
4049          67020  CALL  FMULT
4050          67040  CALL  PUSHF
4051          67060  LXI   H,SINCON
4052          67080  CALL  MUVM
4053          67100  POPR  H    ;MOVE FIRST CONSTANT INTO FAC
4054          67120  MVI   A,4    ;GET X^2
4055          67140  POLY1: PUSH  PSW
4056          67160  PUSHR
4057          67180  PUSH  H    ;SAVE DEGREE
4058          67200  CALL  FMULT
4059          67220  POP  H    ;SAVE CONSTANT POINTER
4060          67240  CALL  MUVRM ;EVALUATE THE POLY, MULTIPLY BY X^2
4061          67260  PUSH  H    ;GET CONSTANT
4062          67280  CALL  FADD
4063          67300  POPR  H    ;MOVE IN CONSTANT
4064          67320  POPR  H    ;MOVE PINTER TO NEXT CONSTANT
4065          67340  POP  PSW   ;GET X^2
4066          67360  DCR   A    ;GET DEGREE
4067          67380  JNZ   POLY1 ;SEE IF DONE
4068          67400  JMP   FMULT> ;IND, DO NEXT TERM
4069          67420  EXTFNC,<
4070  003062* 001000  000041  67440  LXI   H,SINCON ;CALCULATE THE SIN BY EVALUATING
4071  003063* 000000  003100*          ;THE APPROXIMATION POLYNOMIAL
4072  003064* 000000  003662*          ;CONSTANTS FOR SIN, COS
4073  003065* 001000  000003          ;EXTFNC,<
4074  003066* 000000  002613*          ;PI2: 3.141592653589793
4075  003067* 000000  003663*          ;PI2: 3.141592653589793
4076          67460  JMP   POLYX> ; THE APPROXIMATION POLYNOMIAL
4077          67480  IFN    EXTFNC,<
4078  003102* 000000  000233  67500  IFN    EXTFNC,<
4079  003071* 000000  000017  67520  PI2: 3.141592653589793
4080  003072* 000000  008111  67540  017
4081  003073* 000000  000201  67560  111
4082  003074* 000000  000000  67580  201
4083  003075* 000000  000000  67600  FR4: 000  ; 1/4
4084  003076* 000000  000000  67620  000
4085  003077* 000000  000177  67640  000
4086  003108* 000000  000000  67660  177>
4087          67680  SINCON:          ;CONSTANTS FOR SIN, COS
4088  003109* 000000  000005  67700  IFN    EXTFNC,<
4089  003111* 000000  000272  67720  5> ;DEGREE
4090  003112* 000000  000327  67740  272 ; 39,701067
4091  003113* 000000  000036  67760  327
4092  003114* 000000  000000  67780  036
4093  003115* 000000  000000  67800  206
4094  003116* 000000  000000  67820  144 ; -76,57498
4095  003107* 000000  000000  67840  046
4096  003108* 000000  000000  67860  231
4097  003111* 000000  000000  67880  207
4098  003112* 000000  000000  67900  150 ; 81,60223
4099  003113* 000000  000043  67920  064
4100  003114* 000000  000000  67940  043
4101  003115* 000000  000000  67960  207
4102  003116* 000000  000000  67980  548 ; -41,34168
4103  003117* 000000  000045  68000  115
4104  003120* 000000  000226  68020  245
4105  003121* 000000  000232  68040  206
4106  003122* 000000  000017  68060  332 ; 6,283185
4107  003123* 000000  000111  68080  017
4108  003124* 000000  000203  68100  111
4109          68120  203
4110
4111          68180  IFN    EXTFNC,<
4112          68200  TANGENT FUNCTION

```

```

F4      MAC   23=AUG=64 06:08
       SINE, CUSINE AND TANGENT FUNCTIONS

4060          67240  CALL  MUVRM ;GET CONSTANT
4061          67260  PUSH  H    ;SAVE PINTER
4062          67280  CALL  FADD
4063          67300  POPR  H    ;MOVE IN CONSTANT
4064          67320  POPR  H    ;MOVE PINTER TO NEXT CONSTANT
4065          67340  POP  PSW   ;GET X^2
4066          67360  DCR   A    ;GET DEGREE
4067          67380  JNZ   POLY1 ;SEE IF DONE
4068          67400  JMP   FMULT> ;IND, DO NEXT TERM
4069          67420  EXTFNC,<
4070  003062* 001000  000041  67440  LXI   H,SINCON ;CALCULATE THE SIN BY EVALUATING
4071  003063* 000000  003100*          ;THE APPROXIMATION POLYNOMIAL
4072  003064* 000000  003662*          ;CONSTANTS FOR SIN, COS
4073  003065* 001000  000003          ;EXTFNC,<
4074  003066* 000000  002613*          ;PI2: 3.141592653589793
4075  003067* 000000  003663*          ;PI2: 3.141592653589793
4076          67460  JMP   POLYX> ; THE APPROXIMATION POLYNOMIAL
4077          67480  IFN    EXTFNC,<
4078  003102* 000000  000233  67500  IFN    EXTFNC,<
4079  003071* 000000  000017  67520  PI2: 3.141592653589793
4080  003072* 000000  008111  67540  017
4081  003073* 000000  000201  67560  111
4082  003074* 000000  000000  67580  201
4083  003075* 000000  000000  67600  FR4: 000  ; 1/4
4084  003076* 000000  000000  67620  000
4085  003077* 000000  000177  67640  000
4086  003108* 000000  000000  67660  177>
4087          67680  SINCON:          ;CONSTANTS FOR SIN, COS
4088  003109* 000000  000005  67700  IFN    EXTFNC,<
4089  003111* 000000  000272  67720  5> ;DEGREE
4090  003112* 000000  000327  67740  272 ; 39,701067
4091  003113* 000000  000036  67760  327
4092  003114* 000000  000000  67780  036
4093  003115* 000000  000000  67800  206
4094  003116* 000000  000000  67820  144 ; -76,57498
4095  003107* 000000  000000  67840  046
4096  003108* 000000  000000  67860  231
4097  003111* 000000  000000  67880  207
4098  003112* 000000  000000  67900  150 ; 81,60223
4099  003113* 000000  000043  67920  064
4100  003114* 000000  000000  67940  043
4101  003115* 000000  000000  67960  207
4102  003116* 000000  000000  67980  548 ; -41,34168
4103  003117* 000000  000045  68000  115
4104  003120* 000000  000226  68020  245
4105  003121* 000000  000232  68040  206
4106  003122* 000000  000017  68060  332 ; 6,283185
4107  003123* 000000  000111  68080  017
4108  003124* 000000  000203  68100  111
4109          68120  203
4110
4111          68180  IFN    EXTFNC,<
4112          68200  TANGENT FUNCTION

```

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 17-3
F4 MAC 23-AUG-64 06:08 SINE, COSINE AND TANGENT FUNCTIONS

```

4113          68220    ITAN(X)=SIN(X)/COS(X)
4114 003125' 001000 000315
4115 003126' 000000 001205'
4116 003127' 000000 003665'
4117 003128' 000000 001205'
4118 003131' 000000 002770'
4119 003132' 000000 003126'
4120 003135' 001000 000301
4121 003134' 001000 000341
4122 003135' 001000 000305
4123 003139' 000000 001205'
4124 003140' 000000 003131'
4125 003140' 001000 000303
4126 003141' 001000 000315
4127 003142' 000000 001225'
4128 003143' 000000 003130'
4129 003144' 001000 000315
4130 003145' 000000 002764
4131 003146' 000000 003130'
4132 003147' 001000 000303
4133 003150' 000000 000053'
4134 003151' 000000 003145'
4135          68420    PAGE
                                ITAN(X)=SIN(X)/COS(X)
                                CALL      PUSHF      FSAVE ARG
                                CALL      SIN       F TAN(X)=SIN(X)/COS(X)
                                POP      B       JGET X OFF STACK
                                POP      H       JPUHF SMASHES (DE)
                                CALL      PUSHF
                                XCHG
                                CALL      MUVFH      JGET LO'S WHERE THEY BELONG
                                CALL      CUS
                                JMP     FDIVT>

```

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 47(115) 06:09 27-AUG-75 PAGE 18
F4 MAC 23-AUG-64 06:08 ARCTANGENT FUNCTION

```

413b          68440  SUBTTL ARCTANGENT FUNCTION
4137          68450  IFN EXTFNC,<
4138          68460  ;IDEAS USE IDENTITIES TO GET ARG BETWEEN 0 AND 1 AND THEN USE AN
4139          68480  APPROXIMATION POLYNOMIAL TO COMPUTE ARCTAN(X)
4140  003154*  0010000  000357  68520  ATN# FSIGN                ;SEE IF ARG IS NEGATIVE
4141  003154*  0010000  000374  68540  CM PSHNEG                ;IF ARG IS NEGATIVE, USE!
4142  003154*  0000000  002340* 
4143  003155*  0000000  003150* 
4144  003156*  0010000  000374*  68560  CM NEG                 ; ARCTAN(X)==-ARCTAN(-X)
4145  003157*  0000000  001175* 
4146  003158*  0000000  000374* 
4147  003161*  0010000  000672  68580  LDA FAC                 ;SEE IF FAC +GT. 1
4149  003163*  0000000  003157* 
4150  003164*  0010000  000376  68600  CPI 201
4151  003165*  0000000  000201  68620  JC ATN2
4152  003166*  0010000  000332  68640  LXI B,201*400+SCODE ;GET THE CONSTANT 1
4153  003167*  0000000  003295* 
4154  003168*  0010000  000374* 
4155  003170*  0010000  000001  68660  MOV D,C
4156  003172*  0000000  000446*  68680  MOV E,C ;COMPUTE RECIPROCAL TO USE THE IDENTITY:
4157  003173*  0000000  003167* 
4158  003174*  0010000  000121  68700  CALL FDIV ; ARCTAN(X)=PI/2-ARCTAN(1/X)
4159  003175*  0010000  000131 
4160  003176*  0010000  000315 
4161  003177*  0000000  000655* 
4162  003178*  0010000  000320* 
4163  003201*  0010000  000041  68720  LXI H,FSUBS ;PUT FSUBS ON THE STACK SO WE WILL RETURN
4164  003204*  0000000  000111* 
4165  003205*  0000000  003177* 
4166  003204*  0010000  000345  68740  PUSH H ; TO IT AND SUBTRACT THE RESULT FROM PI/2
4167  003205*  0010000  000041  68760  ATN# LXI H,ATNCON ;EVALUATE APPROXIMATION POLYNOMIAL
4168  003206*  0000000  003211* 
4169  003207*  0000000  003202* 
4170  003210*  0010000  000315  68780  CALL POLYX
4171  003211*  0010000  000315* 
4172  003212*  0000000  003208* 
4173  003213*  0010000  000041  68800  LXI H,PI2 ;GET POINTER TO PI/2 IN CASE WE HAVE TO
4174  003214*  0000000  003670* 
4175  003215*  0000000  003211* 
4176  003216*  0010000  000311  68820  RET ; SUBTRACT THE RESULT FROM PI/2
4177
4178          68860  ;CONSTANTS FOR ATN
4179  003217*  0000000  000011  68880  ATNCON: 11 ;DEGREE
4180          68890  000011  000011* 
4181  003221*  0000000  000327  68920  112 ; ,002866226
4182  003222*  0000000  000073  68940  52
4183  003223*  0000000  000170  68960  073
4184  003224*  0000000  000092  68980  002 ; ,01616574
4185  003225*  0000000  000156  69000  156
4186  003226*  0000000  000204  69020  204
4187  003227*  0000000  000173  69040  173
4188  003230*  0000000  000376  69060  376 ; ,04290961

```

MATHPK FOR BASIC MCS 8880 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 18-1
F4 MAC 23-AUG-64 06100 AKCTANGENT FUNCTION

```
4189 003231* 000000 000301 69080 301
4190 003232* 000000 000057 69100 057
4191 003233* 000000 000174 69120 174
4192 003234* 000000 000164 69140 164 ; =,07528964
4193 003235* 000000 000061 69160 061
4194 003236* 000000 000032 69180 232
4195 003237* 000000 000175 69200 175
4196 003240* 000000 000204 69220 204 ; ,1065626
4197 003241* 000000 000075 69240 075
4198 003242* 000000 000132 69260 132
4199 003243* 000000 000175 69280 175
4200 003244* 000000 000310 69300 310 ; =,142089
4201 003245* 000000 000177 69320 177
4202 003246* 000000 000221 69340 221
4203 003247* 000000 000176 69360 176
4204 003250* 000000 000344 69380 344 ; ,1999355
4205 003251* 000000 000273 69400 273
4206 003252* 000000 000114 69420 114
4207 003253* 000000 000176 69440 176
4208 003254* 000000 000154 69460 154 ; =,3333315
4209 003255* 000000 000252 69480 252
4210 003256* 000000 000232 69500 232
4211 003257* 000000 000171 69520 177
4212 003260* 000000 000080 69540 000 ; 1.8
4213 003261* 000000 000000 69560 000
4214 003262* 000000 000000 69580 000
4215 003263* 000000 000201 69600 201>
4216 69620 PAGE
```

MATHPK FOR BASIC MCS 8880 GATES/ALLEN/DAVIDOFF MACRO 47(113) 06:09 27-AUG-75 PAGE 19
F4 MAC 23-AUG-64 06100 SYSTEM INITIALIZATION CODE

```
4217 69640 SUBTTL SYSTEM INITIALIZATION CODE
4218 69660 RADIX 10 ;IN ALL NON-MATH PACKAGE CODE
4219 69680 ;THIS IS THE SYSTEM INITIALIZATION CODE
4220 69700 ;IT SHOULD BE LOADED AT THE END OF THE BASIC
4221 69720 ;INTERPRETER
4222
4223 69760 INTERNAL INIT
4224
4225 69780 EXTERNAL CRD0,LINGET,QINLIN,READY,SCRATCH,STROUT,REASON,BUF
4226 69820 EXTERNAL SNERR,OMERR,ILLFUN
4227
4228 69840 FUNIO=<"0256*#0312>+#040+SCODE
4229 69860 INITSA: BLOCK 10
4230 69880 INITAT: LXI H,AUTXT
4231 69920 CALL STROUT
4232 69940 INIT: IFN REALIO,<
4233 69960 IN 1 ;IGNORE GARBAGE CHARACTER IN INTERFACE
4234 69980 IN "0255 ;SEE WHAT KIND OF I/O HE IS
4235 70000 IN U100 ;IS HE FUNNY TTY?
4236 70020 ANI U100
4237 70040 JZ NOTSIU
4238 69940 IFN 003304* 001000 000335
4239 69960 IN 1
4240 69980 IN "0255
4241 70000 IN U100
4242 70020 ANI U100
4243 70040 JZ NOTSIU
4244 70060 LXI H,FUNIO
4245 70080 SHLD CNLCA2##+2
4246 70100 SHLD CNLCA3##+2
4247 70120 SHLD CNLCA4##+2
4248 70140 IFN LENGTH,<
4249 70160 MVI H,"U304 ;SUBSTITUTE "CNZ"
4250 70180 SHLD CNLCA1##+2
4251 70200 SHLD CNLCA2##+2
4252 70220 SHLD CNLCA3##+2
4253 70240 SHLD CNLCA4##+2
4254 70260 SHLD CNLCA1##+2
4255 70280 SHLD CNLCA2##+2
4256 70300 SHLD CNLCA3##+2
4257 70320 SHLD CNLCA4##+2
4258 70340 SHLD CNLCA1##+2
4259 70360 SHLD CNLCA2##+2
4260 70380 SHLD CNLCA3##+2
4261 70400 SHLD CNLCA4##+2
4262 70420 SHLD CNLCA1##+2
4263 70440 SHLD CNLCA2##+2
4264 70460 SHLD CNLCA3##+2
4265 70480 SHLD CNLCA4##+2
4266 70500 SHLD CNLCA1##+2
4267 70520 SHLD CNLCA2##+2
4268 70540 SHLD CNLCA3##+2
4269 70560 SHLD CNLCA4##+2
4270 70580 SHLD CNLCA1##+2
4271 70600 SHLD CNLCA2##+2
4272 70620 SHLD CNLCA3##+2
4273 70640 SHLD CNLCA4##+2
4274 70660 SHLD CNLCA1##+2
4275 70680 SHLD CNLCA2##+2
4276 70700 SHLD CNLCA3##+2
4277 70720 SHLD CNLCA4##+2
4278 70740 SHLD CNLCA1##+2
4279 70760 SHLD CNLCA2##+2
4280 70780 SHLD CNLCA3##+2
4281 70800 SHLD CNLCA4##+2
4282 70820 SHLD CNLCA1##+2
4283 70840 SHLD CNLCA2##+2
4284 70860 SHLD CNLCA3##+2
4285 70880 SHLD CNLCA4##+2
4286 70900 SHLD CNLCA1##+2
4287 70920 SHLD CNLCA2##+2
4288 70940 SHLD CNLCA3##+2
4289 70960 SHLD CNLCA4##+2
4290 70980 SHLD CNLCA1##+2
4291 71000 SHLD CNLCA2##+2
4292 71020 SHLD CNLCA3##+2
4293 71040 SHLD CNLCA4##+2
4294 71060 SHLD CNLCA1##+2
4295 71080 SHLD CNLCA2##+2
4296 71100 SHLD CNLCA3##+2
4297 71120 SHLD CNLCA4##+2
4298 71140 SHLD CNLCA1##+2
4299 71160 SHLD CNLCA2##+2
4300 71180 SHLD CNLCA3##+2
4301 71200 SHLD CNLCA4##+2
4302 71220 SHLD CNLCA1##+2
4303 71240 SHLD CNLCA2##+2
4304 71260 SHLD CNLCA3##+2
4305 71280 SHLD CNLCA4##+2
4306 71300 SHLD CNLCA1##+2
4307 71320 SHLD CNLCA2##+2
4308 71340 SHLD CNLCA3##+2
4309 71360 SHLD CNLCA4##+2
4310 71380 SHLD CNLCA1##+2
4311 71400 SHLD CNLCA2##+2
4312 71420 SHLD CNLCA3##+2
4313 71440 SHLD CNLCA4##+2
4314 71460 SHLD CNLCA1##+2
4315 71480 SHLD CNLCA2##+2
4316 71500 SHLD CNLCA3##+2
4317 71520 SHLD CNLCA4##+2
4318 71540 SHLD CNLCA1##+2
4319 71560 SHLD CNLCA2##+2
4320 71580 SHLD CNLCA3##+2
4321 71600 SHLD CNLCA4##+2
4322 71620 SHLD CNLCA1##+2
4323 71640 SHLD CNLCA2##+2
4324 71660 SHLD CNLCA3##+2
4325 71680 SHLD CNLCA4##+2
4326 71700 SHLD CNLCA1##+2
4327 71720 SHLD CNLCA2##+2
4328 71740 SHLD CNLCA3##+2
4329 71760 SHLD CNLCA4##+2
4330 71780 SHLD CNLCA1##+2
4331 71800 SHLD CNLCA2##+2
4332 71820 SHLD CNLCA3##+2
4333 71840 SHLD CNLCA4##+2
4334 71860 SHLD CNLCA1##+2
4335 71880 SHLD CNLCA2##+2
4336 71900 SHLD CNLCA3##+2
4337 71920 SHLD CNLCA4##+2
4338 71940 SHLD CNLCA1##+2
4339 71960 SHLD CNLCA2##+2
4340 71980 SHLD CNLCA3##+2
4341 72000 SHLD CNLCA4##+2
4342 72020 SHLD CNLCA1##+2
4343 72040 SHLD CNLCA2##+2
4344 72060 SHLD CNLCA3##+2
4345 72080 SHLD CNLCA4##+2
4346 72100 SHLD CNLCA1##+2
4347 72120 SHLD CNLCA2##+2
4348 72140 SHLD CNLCA3##+2
4349 72160 SHLD CNLCA4##+2
4350 72180 SHLD CNLCA1##+2
4351 72200 SHLD CNLCA2##+2
4352 72220 SHLD CNLCA3##+2
4353 72240 SHLD CNLCA4##+2
4354 72260 SHLD CNLCA1##+2
4355 72280 SHLD CNLCA2##+2
4356 72300 SHLD CNLCA3##+2
4357 72320 SHLD CNLCA4##+2
4358 72340 SHLD CNLCA1##+2
4359 72360 SHLD CNLCA2##+2
4360 72380 SHLD CNLCA3##+2
4361 72400 SHLD CNLCA4##+2
4362 72420 SHLD CNLCA1##+2
4363 72440 SHLD CNLCA2##+2
4364 72460 SHLD CNLCA3##+2
4365 72480 SHLD CNLCA4##+2
4366 72500 SHLD CNLCA1##+2
4367 72520 SHLD CNLCA2##+2
4368 72540 SHLD CNLCA3##+2
4369 72560 SHLD CNLCA4##+2
4370 72580 SHLD CNLCA1##+2
4371 72600 SHLD CNLCA2##+2
4372 72620 SHLD CNLCA3##+2
4373 72640 SHLD CNLCA4##+2
4374 72660 SHLD CNLCA1##+2
4375 72680 SHLD CNLCA2##+2
4376 72700 SHLD CNLCA3##+2
4377 72720 SHLD CNLCA4##+2
4378 72740 SHLD CNLCA1##+2
4379 72760 SHLD CNLCA2##+2
4380 72780 SHLD CNLCA3##+2
4381 72800 SHLD CNLCA4##+2
4382 72820 SHLD CNLCA1##+2
4383 72840 SHLD CNLCA2##+2
4384 72860 SHLD CNLCA3##+2
4385 72880 SHLD CNLCA4##+2
4386 72900 SHLD CNLCA1##+2
4387 72920 SHLD CNLCA2##+2
4388 72940 SHLD CNLCA3##+2
4389 72960 SHLD CNLCA4##+2
4390 72980 SHLD CNLCA1##+2
4391 73000 SHLD CNLCA2##+2
4392 73020 SHLD CNLCA3##+2
4393 73040 SHLD CNLCA4##+2
4394 73060 SHLD CNLCA1##+2
4395 73080 SHLD CNLCA2##+2
4396 73100 SHLD CNLCA3##+2
4397 73120 SHLD CNLCA4##+2
4398 73140 SHLD CNLCA1##+2
4399 73160 SHLD CNLCA2##+2
4400 73180 SHLD CNLCA3##+2
4401 73200 SHLD CNLCA4##+2
4402 73220 SHLD CNLCA1##+2
4403 73240 SHLD CNLCA2##+2
4404 73260 SHLD CNLCA3##+2
4405 73280 SHLD CNLCA4##+2
4406 73300 SHLD CNLCA1##+2
4407 73320 SHLD CNLCA2##+2
4408 73340 SHLD CNLCA3##+2
4409 73360 SHLD CNLCA4##+2
4410 73380 SHLD CNLCA1##+2
4411 73400 SHLD CNLCA2##+2
4412 73420 SHLD CNLCA3##+2
4413 73440 SHLD CNLCA4##+2
4414 73460 SHLD CNLCA1##+2
4415 73480 SHLD CNLCA2##+2
4416 73500 SHLD CNLCA3##+2
4417 73520 SHLD CNLCA4##+2
4418 73540 SHLD CNLCA1##+2
4419 73560 SHLD CNLCA2##+2
4420 73580 SHLD CNLCA3##+2
4421 73600 SHLD CNLCA4##+2
4422 73620 SHLD CNLCA1##+2
4423 73640 SHLD CNLCA2##+2
4424 73660 SHLD CNLCA3##+2
4425 73680 SHLD CNLCA4##+2
4426 73700 SHLD CNLCA1##+2
4427 73720 SHLD CNLCA2##+2
4428 73740 SHLD CNLCA3##+2
4429 73760 SHLD CNLCA4##+2
4430 73780 SHLD CNLCA1##+2
4431 73800 SHLD CNLCA2##+2
4432 73820 SHLD CNLCA3##+2
4433 73840 SHLD CNLCA4##+2
4434 73860 SHLD CNLCA1##+2
4435 73880 SHLD CNLCA2##+2
4436 73900 SHLD CNLCA3##+2
4437 73920 SHLD CNLCA4##+2
4438 73940 SHLD CNLCA1##+2
4439 73960 SHLD CNLCA2##+2
4440 73980 SHLD CNLCA3##+2
4441 74000 SHLD CNLCA4##+2
4442 74020 SHLD CNLCA1##+2
4443 74040 SHLD CNLCA2##+2
4444 74060 SHLD CNLCA3##+2
4445 74080 SHLD CNLCA4##+2
4446 74100 SHLD CNLCA1##+2
4447 74120 SHLD CNLCA2##+2
4448 74140 SHLD CNLCA3##+2
4449 74160 SHLD CNLCA4##+2
4450 74180 SHLD CNLCA1##+2
4451 74200 SHLD CNLCA2##+2
4452 74220 SHLD CNLCA3##+2
4453 74240 SHLD CNLCA4##+2
4454 74260 SHLD CNLCA1##+2
4455 74280 SHLD CNLCA2##+2
4456 74300 SHLD CNLCA3##+2
4457 74320 SHLD CNLCA4##+2
4458 74340 SHLD CNLCA1##+2
4459 74360 SHLD CNLCA2##+2
4460 74380 SHLD CNLCA3##+2
4461 74400 SHLD CNLCA4##+2
4462 74420 SHLD CNLCA1##+2
4463 74440 SHLD CNLCA2##+2
4464 74460 SHLD CNLCA3##+2
4465 74480 SHLD CNLCA4##+2
4466 74500 SHLD CNLCA1##+2
4467 74520 SHLD CNLCA2##+2
4468 74540 SHLD CNLCA3##+2
4469 74560 SHLD CNLCA4##+2
4470 74580 SHLD CNLCA1##+2
4471 74600 SHLD CNLCA2##+2
4472 74620 SHLD CNLCA3##+2
4473 74640 SHLD CNLCA4##+2
4474 74660 SHLD CNLCA1##+2
4475 74680 SHLD CNLCA2##+2
4476 74700 SHLD CNLCA3##+2
4477 74720 SHLD CNLCA4##+2
4478 74740 SHLD CNLCA1##+2
4479 74760 SHLD CNLCA2##+2
4480 74780 SHLD CNLCA3##+2
4481 74800 SHLD CNLCA4##+2
4482 74820 SHLD CNLCA1##+2
4483 74840 SHLD CNLCA2##+2
4484 74860 SHLD CNLCA3##+2
4485 74880 SHLD CNLCA4##+2
4486 74900 SHLD CNLCA1##+2
4487 74920 SHLD CNLCA2##+2
4488 74940 SHLD CNLCA3##+2
4489 74960 SHLD CNLCA4##+2
4490 74980 SHLD CNLCA1##+2
4491 75000 SHLD CNLCA2##+2
4492 75020 SHLD CNLCA3##+2
4493 75040 SHLD CNLCA4##+2
4494 75060 SHLD CNLCA1##+2
4495 75080 SHLD CNLCA2##+2
4496 75100 SHLD CNLCA3##+2
4497 75120 SHLD CNLCA4##+2
4498 75140 SHLD CNLCA1##+2
4499 75160 SHLD CNLCA2##+2
4500 75180 SHLD CNLCA3##+2
4501 75200 SHLD CNLCA4##+2
4502 75220 SHLD CNLCA1##+2
4503 75240 SHLD CNLCA2##+2
4504 75260 SHLD CNLCA3##+2
4505 75280 SHLD CNLCA4##+2
4506 75300 SHLD CNLCA1##+2
4507 75320 SHLD CNLCA2##+2
4508 75340 SHLD CNLCA3##+2
4509 75360 SHLD CNLCA4##+2
4510 75380 SHLD CNLCA1##+2
4511 75400 SHLD CNLCA2##+2
4512 75420 SHLD CNLCA3##+2
4513 75440 SHLD CNLCA4##+2
4514 75460 SHLD CNLCA1##+2
4515 75480 SHLD CNLCA2##+2
4516 75500 SHLD CNLCA3##+2
4517 75520 SHLD CNLCA4##+2
4518 75540 SHLD CNLCA1##+2
4519 75560 SHLD CNLCA2##+2
4520 75580 SHLD CNLCA3##+2
4521 75600 SHLD CNLCA4##+2
4522 75620 SHLD CNLCA1##+2
4523 75640 SHLD CNLCA2##+2
4524 75660 SHLD CNLCA3##+2
4525 75680 SHLD CNLCA4##+2
4526 75700 SHLD CNLCA1##+2
4527 75720 SHLD CNLCA2##+2
4528 75740 SHLD CNLCA3##+2
4529 75760 SHLD CNLCA4##+2
4530 75780 SHLD CNLCA1##+2
4531 75800 SHLD CNLCA2##+2
4532 75820 SHLD CNLCA3##+2
4533 75840 SHLD CNLCA4##+2
4534 75860 SHLD CNLCA1##+2
4535 75880 SHLD CNLCA2##+2
4536 75900 SHLD CNLCA3##+2
4537 75920 SHLD CNLCA4##+2
4538 75940 SHLD CNLCA1##+2
4539 75960 SHLD CNLCA2##+2
4540 75980 SHLD CNLCA3##+2
4541 76000 SHLD CNLCA4##+2
4542 76020 SHLD CNLCA1##+2
4543 76040 SHLD CNLCA2##+2
4544 76060 SHLD CNLCA3##+2
4545 76080 SHLD CNLCA4##+2
4546 76100 SHLD CNLCA1##+2
4547 76120 SHLD CNLCA2##+2
4548 76140 SHLD CNLCA3##+2
4549 76160 SHLD CNLCA4##+2
4550 76180 SHLD CNLCA1##+2
4551 76200 SHLD CNLCA2##+2
4552 76220 SHLD CNLCA3##+2
4553 76240 SHLD CNLCA4##+2
4554 76260 SHLD CNLCA1##+2
4555 76280 SHLD CNLCA2##+2
4556 76300 SHLD CNLCA3##+2
4557 76320 SHLD CNLCA4##+2
4558 76340 SHLD CNLCA1##+2
4559 76360 SHLD CNLCA2##+2
4560 76380 SHLD CNLCA3##+2
4561 76400 SHLD CNLCA4##+2
4562 76420 SHLD CNLCA1##+2
4563 76440 SHLD CNLCA2##+2
4564 76460 SHLD CNLCA3##+2
4565 76480 SHLD CNLCA4##+2
4566 76500 SHLD CNLCA1##+2
4567 76520 SHLD CNLCA2##+2
4568 76540 SHLD CNLCA3##+2
4569 76560 SHLD CNLCA4##+2
4570 76580 SHLD CNLCA1##+2
4571 76600 SHLD CNLCA2##+2
4572 76620 SHLD CNLCA3##+2
4573 76640 SHLD CNLCA4##+2
4574 76660 SHLD CNLCA1##+2
4575 76680 SHLD CNLCA2##+2
4576 76700 SHLD CNLCA3##+2
4577 76720 SHLD CNLCA4##+2
4578 76740 SHLD CNLCA1##+2
4579 76760 SHLD CNLCA2##+2
4580 76780 SHLD CNLCA3##+2
4581 76800 SHLD CNLCA4##+2
4582 76820 SHLD CNLCA1##+2
4583 76840 SHLD CNLCA2##+2
4584 76860 SHLD CNLCA3##+2
4585 76880 SHLD CNLCA4##+2
4586 76900 SHLD CNLCA1##+2
4587 76920 SHLD CNLCA2##+2
4588 76940 SHLD CNLCA3##+2
4589 76960 SHLD CNLCA4##+2
4590 76980 SHLD CNLCA1##+2
4591 77000 SHLD CNLCA2##+2
4592 77020 SHLD CNLCA3##+2
4593 77040 SHLD CNLCA4##+2
4594 77060 SHLD CNLCA1##+2
4595 77080 SHLD CNLCA2##+2
4596 77100 SHLD CNLCA3##+2
4597 77120 SHLD CNLCA4##+2
4598 77140 SHLD CNLCA1##+2
4599 77160 SHLD CNLCA2##+2
4600 77180 SHLD CNLCA3##+2
4601 77200 SHLD CNLCA4##+2
4602 77220 SHLD CNLCA1##+2
4603 77240 SHLD CNLCA2##+2
4604 77260 SHLD CNLCA3##+2
4605 77280 SHLD CNLCA4##+2
4606 77300 SHLD CNLCA1##+2
4607 77320 SHLD CNLCA2##+2
4608 77340 SHLD CNLCA3##+2
4609 77360 SHLD CNLCA4##+2
4610 77380 SHLD CNLCA1##+2
4611 77400 SHLD CNLCA2##+2
4612 77420 SHLD CNLCA3##+2
4613 77440 SHLD CNLCA4##+2
4614 77460 SHLD CNLCA1##+2
4615 77480 SHLD CNLCA2##+2
4616 77500 SHLD CNLCA3##+2
4617 77520 SHLD CNLCA4##+2
4618 77540 SHLD CNLCA1##+2
4619 77560 SHLD CNLCA2##+2
4620 77580 SHLD CNLCA3##+2
4621 77600 SHLD CNLCA4##+2
4622 77620 SHLD CNLCA1##+2
4623 77640 SHLD CNLCA2##+2
4624 77660 SHLD CNLCA3##+2
4625 77680 SHLD CNLCA4##+2
4626 77700 SHLD CNLCA1##+2
4627 77720 SHLD CNLCA2##+2
4628 77740 SHLD CNLCA3##+2
4629 77760 SHLD CNLCA4##+2
4630 77780 SHLD CNLCA1##+2
4631 77800 SHLD CNLCA2##+2
4632 77820 SHLD CNLCA3##+2
4633 77840 SHLD CNLCA4##+2
4634 77860 SHLD CNLCA1##+2
4635 77880 SHLD CNLCA2##+2
4636 77900 SHLD CNLCA3##+2
4637 77920 SHLD CNLCA4##+2
4638 77940 SHLD CNLCA1##+2
4639 77960 SHLD CNLCA2##+2
4640 77980 SHLD CNLCA3##+2
4641 78000 SHLD CNLCA4##+2
4642 78020 SHLD CNLCA1##+2
4643 78040 SHLD CNLCA2##+2
4644 78060 SHLD CNLCA3##+2
4645 78080 SHLD CNLCA4##+2
4646 78100 SHLD CNLCA1##+2
4647 78120 SHLD CNLCA2##+2
4648 78140 SHLD CNLCA3##+2
4649 78160 SHLD CNLCA4##+2
4650 78180 SHLD CNLCA1##+2
4651 78200 SHLD CNLCA2##+2
4652 78220 SHLD CNLCA3##+2
4653 78240 SHLD CNLCA4##+2
4654 78260 SHLD CNLCA1##+2
4655 78280 SHLD CNLCA2##+2
4656 78300 SHLD CNLCA3##+2
4657 78320 SHLD CNLCA4##+2
4658 78340 SHLD CNLCA1##+2
4659 78360 SHLD CNLCA2##+2
4660 78380 SHLD CNLCA3##+2
4661 78400 SHLD CNLCA4##+2
4662 78420 SHLD CNLCA1##+2
4663 78440 SHLD CNLCA2##+2
4664 78460 SHLD CNLCA3##+2
4665 78480 SHLD CNLCA4##+2
4666 78500 SHLD CNLCA1##+2
4667 78520 SHLD CNLCA2##+2
4668 78540 SHLD CNLCA3##+2
4669 78560 SHLD CNLCA4##+2
4670 78580 SHLD CNLCA1##+2
4671 78600 SHLD CNLCA2##+2
4672 78620 SHLD CNLCA3##+2
4673 78640 SHLD CNLCA4##+2
4674 78660 SHLD CNLCA1##+2
4675 78680 SHLD CNLCA2##+2
4676 78700 SHLD CNLCA3##+2
4677 78720 SHLD CNLCA4##+2
4678 78740 SHLD CNLCA1##+2
4679 78760 SHLD CNLCA2##+2
4680 78780 SHLD CNLCA3##+2
4681 78800 SHLD CNLCA4##+2
4682 78820 SHLD CNLCA1##+2
4683 78840 SHLD CNLCA2##+2
4684 78860 SHLD CNLCA3##+2
4685 78880 SHLD CNLCA4##+2
4686 78900 SHLD CNLCA1##+2
4687 78920 SHLD CNLCA2##+2
4688 78940 SHLD CNLCA3##+2
4689 78960 SHLD CNLCA4##+2
4690 78980 SHLD CNLCA1##+2
4691 79000 SHLD CNLCA2##+2
4692 79020 SHLD CNLCA3##+2
4693 79040 SHLD CNLCA4##+2
4694 79060 SHLD CNLCA1##+2
4695 79080 SHLD CNLCA2##+2
4696 79100 SHLD CNLCA3##+2
4697 79120 SHLD CNLCA4##+2
4698 79140 SHLD CNLCA1##+2
4699 79160 SHLD CNLCA2##+2
4700 79180 SHLD CNLCA3##+2
4701 79200 SHLD CNLCA4##+2
4702 79220 SHLD CNLCA1##+2
4703 79240 SHLD CNLCA2##+2
4704 79260 SHLD CNLCA3##+2
4705 79280 SHLD CNLCA4##+2
4706 79300 SHLD CNLCA1##+2
4707 79320 SHLD CNLCA2##+2
4708 79340 SHLD CNLCA3##+2
4709 79360 SHLD CNLCA4##+2
4710 79380 SHLD CNLCA1##+2
4711 79400 SHLD CNLCA2##+2
4712 79420 SHLD CNLCA3##+2
4713 79440 SHLD CNLCA4##+2
4714 79460 SHLD CNLCA1##+2
4715 79480 SHLD CNLCA2##+2
4716 79500 SHLD CNLCA3##+2
4717 79520 SHLD CNLCA4##+2
4718 79540 SHLD CNLCA1##+2
4719 79560 SHLD CNLCA2##+2
4720 79580 SHLD CNLCA3##+2
4721 79600 SHLD CNLCA4##+2
4722 79620 SHLD CNLCA1##+2
4723 79640 SHLD CNLCA2##+2
4724 79660 SHLD CNLCA3##+2
4725 79680 SHLD CNLCA4##+2
4726 79700 SHLD CNLCA1##+2
4727 79720 SHLD CNLCA2##+2
4728 79740 SHLD CNLCA3##+2
4729 79760 SHLD CNLCA4##+2
4730 79780 SHLD CNLCA1##+2
4731 79800 SHLD CNLCA2##+2
4732 79820 SHLD CNLCA3##+2
4733 79840 SHLD CNLCA4##+2
4734 79860 SHLD CNLCA1##+2
4735 79880 SHLD CNLCA2##+2
4736 79900 SHLD CNLCA3##+2
4737 79920 SHLD CNLCA4##+2
4738 79940 SHLD CNLCA1##+2
4739 79960 SHLD CNLCA2##+2
4740 79980 SHLD CNLCA3##+2
4741 80000 SHLD CNLCA4##+2
4742 80020 SHLD CNLCA1##+2
4743 80040 SHLD CNLCA2##+2
4744 80060 SHLD CNLCA3##+2
4745 80080 SHLD CNLCA4##+2
4746 80100 SHLD CNLCA1##+2
4747 80120 SHLD CNLCA2##+2
4748 80140 SHLD CNLCA3##+2
4749 80160 SHLD CNLCA4##+2
4750 80180 SHLD CNLCA1##+2
4751 80200 SHLD CNLCA2##+2
4752 80220 SHLD CNLCA3##+2
4753 80240 SHLD CNLCA4##+2
4754 80260 SHLD CNLCA1##+2
4755 80280 SHLD CNLCA2##+2
4756 80300 SHLD CNLCA3##+2
4757 80320 SHLD CNLCA4##+2
4758 80340 SHLD CNLCA1##+2
4759 80360 SHLD CNLCA2##+2
4760 80380 SHLD CNLCA3##+2
4761 80400 SHLD CNLCA4##+2
4762 80420 SHLD CNLCA1##+2
4763 80440 SHLD CNLCA2##+2
4764 80460 SHLD CNLCA3##+2
4765 80480 SHLD CNLCA4##+2
4766 80500 SHLD CNLCA1##+2
4767 80520 SHLD CNLCA2##+2
4768 80540 SHLD CNLCA3##+2
4769 80560 SHLD CNLCA4##+2
4770 80580 SHLD CNLCA1##+2
4771 80
```

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACHO 47{113} 06:09 27-AUG-75 PAGE 19-1

F4 MAC 23-AUG-64 06108 SYSTEM INITIALIZATION CODE

4270 003342* 001000 000000 70650 NUTSIO: IN *D255
4271 003343* 001000 000333 70660 ANI *040
4272 003344* 000000 000377 70670
4273 003345* 001000 000346 70680
4274 003346* 000000 000046 70690
4275 003347* 001000 000312 70700 JZ NOTPIO
4276 003348* 000000 000340*
4277 003351* 000000 000341*
4278 145002 70720 FUNIO==SCODE+<"0256*0312>+2
4279 003352* 001000 000041 70730 LXI H,FUNIO
4280 003353* 000000 145002*
4281 003354* 000000 0003350*
4282 003355* 001000 000042 707360 SHLD CNLCA2+2
4283 003356* 000000 000043 707400 SHLD CNLCA3+2
4284 003357* 000000 000351*
4285 003360* 001000 000045 707480 MVI H,70310
4286 003361* 000000 000510 707500
4287 003362* 001000 000042 707400 SHLD CNLCA3+2
4288 003363* 000000 000002*
4289 003364* 000000 0003350*
4290 145001 707420 IFN LENGTH,<
4291 003365* 001000 000046 707440 MVI H,0304
4292 003366* 000000 000254 707460 SHLD CNLCA4+2
4293 003367* 001000 000042 707480
4294 003370* 000000 000002*
4295 003371* 000000 003363*
4296 145001 707480 FUNIO==SCODE+<"0256*0312>+1
4297 003372* 001000 000041 707500 LXI H,FUNIO
4298 003373* 000000 145001*
4299 003374* 001000 0003370*
4300 003375* 000000 000442 707520 SHLD CNLCA1+2
4301 003376* 000000 000044*
4302 003377* 000000 003373*
4303 003400* 707540 NOTPIO:>
4304 003400* 001000 000041 707560 LXI H,SCODE+065535
4305 003401* 000000 177777*
4306 003402* 000000 003376*
4307 003403* 001000 000042 707580 SHLD CURLIN#> FIN CASE OF ERROR MESSAGE
4308 003404* 000000 000004*
4309 003405* 000000 003401*
4310 003406* 001000 000041 707600 LXI H,TSTACK SET UP TEMP STACK
4311 003407* 000000 004315*
4312 003410* 000000 003404*
4313 003411* 001000 000371 707620 SPML
4314 003412* 001000 000042 707640 SHLD STKTOP#>
4315 003413* 000000 000000*
4316 003414* 000000 003407*
4317 707660 IFN CNTTRW,
4318 003415* 001000 000257 707680 XRA A
4319 003416* 001000 000062 707700 STA CNTHFL#>
4320 003417* 000000 000000*
4321 003420* 000000 003413*
4322 003421* 001000 000315 707720 CALL CRDU I TYPE A CR

MATHPK FOR BASIC MCS 8880 GATES/ALLEN/DAVIDOFF MACHO 47{113} 06:09 27-AUG-75 PAGE 19-2

F4 MAC 23-AUG-64 06108 SYSTEM INITIALIZATION CODE

4323 003422* 000000 000002*
4324 003423* 000000 003417*
4325 003424* 001000 000041 707840 IFN STHNG,<
4326 003425* 000000 000000* 707860 LXI H,TEMPS#>
4327 003426* 000000 003422*
4328 003427* 001000 000042 707880 SHLD TEMPP#>
4329 003430* 000000 000000*
4330 003431* 000000 003425*
4331 003432* 000000 000000*
4332 003424* 001000 000041 707880 IFN REALIO,<
4333 003425* 000000 004243* 707900 LXI H,MEMORY TASK HOW MUCH MEMORY AVAILABLE
4334 003426* 000000 003430*
4335 003427* 000000 003431*
4336 003433* 001000 0000315 707840 CALL STROUT SCALL THE STRING PRINTER
4337 003435* 000000 005302*
4338 003437* 000000 003433*
4339 003440* 001000 000315 707860 CALL QINLIN IGET A LINE OF INPUT
4340 003441* 000000 000002*
4341 003442* 000000 003434*
4342 003443* 001000 000327 707880 CHRGRET CPI RAM IGET A CHAR
4343 003444* 001000 0003376 707900 JNZ USEDEF9 INON ZERO, DONT USE DEFAULT
4344 003445* 000000 000100 707920 JZ INITAT
4345 003446* 001000 000312 707940 ORA A
4346 003447* 000000 003276* 707960 JNZ USEDEF9
4347 003450* 000000 003441*
4348 003451* 001000 000267 707980 DRA A
4349 003452* 001000 0003024 707990 JNZ USEDEF9
4350 003453* 000000 003441*
4351 003454* 000000 003447*
4352 003455* 001000 000041 707980 LXI H,LASTHR
4353 003456* 000000 004257*
4354 003457* 000000 003453*
4355 003460* 001000 000043 71000 LOOPMM: INX H
4356 003461* 001000 000076 71020 MVI A,311
4357 003462* 000000 0000467
4358 003463* 001000 000057 71040 MOV M,A
4359 003464* 000000 0000575 71060 CMP M
4360 003465* 001000 000392 71080 JNZ USEDEF9
4361 003466* 000000 003515*
4362 003467* 000000 003456*
4363 003470* 001000 000075 71100 DCR A
4364 003471* 001000 0000167 71120 MOV M,A
4365 003472* 001000 000276 71140 CMP M
4366 003473* 000000 000512 71160 JZ LOOPMM
4367 003474* 000000 003515*
4368 003475* 000000 003466*
4369 003476* 001000 000363 71180 JMP USEDEF9
4370 003477* 000000 003515*
4371 003500* 000000 003474*
4372 003501* 001000 000041 71200 USEDEF9: LXI H,BUF
4373 003502* 000000 000000*
4374 003503* 000000 003477*
4375 003504* 001000 000315 71220 CALL LINGET IGET DECIMAL AMOUNT OF MEMORY IN [D,E]

```

F4 MAC 23=AUG=64 06108 SYSTEM INITIALIZATION CODE

4376 003505* 000000 000000* 71240 ORA A
4377 003506* 000000 003502* 71260 JNZ SNERR ;MAKE SURE HE HAS A TERMINATOR
4378 003507* 001000 000267
4379 003510* 001000 000302*
4380 003511* 001000 000304*
4381 003512* 001000 003529*
4382 003513* 001000 000853 71260 XCHG
4383 003514* 001000 000853 71300 DCX H
4384 003515* 001000 000053 71320 USEDEF: DCX H>
4385 003516* 001000 003529* 71340 IFE REAL10,<
4386 003517* 001000 000853 71360 LXI H,SCODE+16190>
4387 003518* 001000 000845 71380 PUSH H ;ALSO SAVE FOR LATER
4388 003519* 001000 000845 71400 TTYN: LXI H,TTYNH
4389 003520* 001000 000841* 71420 CALL STROUT
4390 003521* 000000 003511* 71440 CALL QINLIN
4391 003522* 001000 000815
4392 003523* 000000 003436*
4393 003524* 000000 003529*
4394 003525* 001000 000815 71440 CALL QINLIN
4395 003526* 000000 000841*
4396 003527* 000000 000841* 71460 CHRGET
4397 003528* 001000 000847 71480 ORA A
4398 003529* 001000 000847 71500 JZ DFLENT
4400 003530* 000000 003604*
4401 003534* 000000 003526*
4402 003535* 001000 000841 71520 LXI H,BUF
4403 003536* 000000 003502*
4404 003537* 000000 003533* 71540 CALL LINGET
4405 003538* 000000 003515* 71560 MOV A,D
4406 003541* 000000 003505* 71580 MOV A,E
4407 003542* 000000 003530* 71600 CPI 16
4408 003543* 001000 000172 71620 JC TTYN
4409 003544* 001000 000267 71640 ADD E
4410 003545* 001000 000802 71660 STA LINPT1## ;DECLARE LINPT1 EXTERNAL
4411 003546* 000000 003517*
4412 003547* 000000 003541* 71680 STA LINPT1## ;DECLARE LINPT1 EXTERNAL
4413 003548* 001000 000802 71700 MOV A,E
4414 003551* 001000 000876 71720 CPI 16
4415 003552* 000000 000620 71740 JC TTYN
4416 003553* 001000 000832 71760 ADD E
4417 003554* 000000 003517* 71780 STA LINPT2##>
4418 003555* 000000 003546* 71800 STA LINPT2##>
4419 003556* 001000 000862 71820 STA LINPT3##>
4420 003557* 000000 000800* 71840 STA LINPT3##>
4421 003558* 000000 003554* 71860 STA LINPT3##>
4422 003559* 001000 000862 71880 IFN LENGTH,<
4423 003561* 001000 000862 71900 STA LINPT2##>
4424 003562* 000000 000000* 71920 IFN STRING,<
4425 003563* 000000 003557* 71940 STA LINPT4##>
4426 003564* 000000 003554* 71960 DFLENT:
4427 003564* 001000 000000 71980 IFN STRING,< ;SET UP DEFAULT STRING SPACE
4428 003565* 000000 000000* 71980 LXI D,SCODE+"05536-"D$0*1

```

```

F4 MAC 23=AUG=64 06108 SYSTEM INITIALIZATION CODE

4429 003566* 000000 003562* 71780 MURCPS: SUI 14
4430 003567* 001000 000326
4431 003570* 000000 000016 71800 JNC MURCPS
4432 003571* 001000 000322
4433 003572* 000000 003517* 71820 ADI 28
4434 003573* 000000 003565* 71840 CMA
4435 003574* 001000 000830 71860 INR A
4436 003575* 000000 000034 71880 ADD E
4437 003576* 001000 000057 71900 STA LINPT4##>
4438 003577* 001000 000074 71920 DFLENT:
4439 003580* 001000 000203 71940 IFN STRING,< ;SET UP DEFAULT STRING SPACE
4440 003581* 001000 000662 71960 LXI D,SCODE+"05536-"D$0*1
4441 003582* 000000 000000* 71980 SHLD MEMSIZE## ;SAVE IN REAL MEMORY SIZE
4442 003583* 000000 003572* 72000 SHLD FRETUP## ;STRINGS START FROM HERE DOWN
4443 003584* 000000 003572* 72020 SHLD FRETUP## ;STRINGS START FROM HERE DOWN
4444 003585* 001000 000021 72040 DAD D ;CALC STRTOP BY SUBTRACTING 200 FROM STKTOP
4445 003586* 000000 003502* 72060 JNC OMERR ;MUST BE POSITIVE
4446 003587* 000000 177717* 72080 POP H
4447 003588* 001000 000341 72080 SHLD MEMSIZE## ;SAVE IN REAL MEMORY SIZE
4448 003589* 001000 000042 72080 POP H
4449 003590* 000000 000042 72080 SHLD MEMSIZE## ;SAVE IN REAL MEMORY SIZE
4450 003591* 000000 000000* 72080 SHLD FRETUP## ;STRINGS START FROM HERE DOWN
4451 003592* 000000 003505* 72080 SHLD FRETUP## ;STRINGS START FROM HERE DOWN
4452 003593* 001000 000042 72080 SHLD FRETUP## ;STRINGS START FROM HERE DOWN
4453 003594* 000000 000000* 72080 SHLD FRETUP## ;STRINGS START FROM HERE DOWN
4454 003595* 000000 003511* 72080 SHLD FRETUP## ;STRINGS START FROM HERE DOWN
4455 003596* 001000 000051 72080 DAD D ;CALC STRTOP BY SUBTRACTING 200 FROM STKTOP
4456 003597* 001000 000322 72080 JNC OMERR ;MUST BE POSITIVE
4457 003598* 000000 000000* 72080 POP H ;PUSH IT ON STACK
4458 003599* 000000 003519* 72080 POP H ;PUSH IT ON STACK
4459 003602* 001000 000053 72080 DCX H ;MOVE LOWER IS STKTOP
4460 003603* 001000 000349 72080 PUSH H ;SAVE IT ON STACK
4461 72120 IFE EXTFCN,<
4462 72140 ; FUNCTION DELETION ROUTINE FOR 4K MACHINE
4463 72160 ; TO ADD A NEW FUNCTION JUST UPDATE TBLDD
4464 72180 ; ASKAGN: LXI H,TBLDD ;START OF FUNCTION TABLE
4465 72200 LUPASK: PUSH H ;PUT ON CANDIDATE FOR START OF FREE MEMORY
4466 72240 LXI D,TBDAWK ;PUT ON MESSAGE LOCATION
4467 72280 COMPAR ;(H,L)=MESSAGE POINTER
4468 72300 JZ FINFUN ;YES, GET FIRST FREE MEM LOC
4469 72320 JINTO D,E1 AND QUIT
4470 72340 PUSHM ;PUT ON MESSAGE LOCATION
4471 72360 XTHL ;(H,L)=MESSAGE POINTER
4472 72380 CALL STROUT ;POINT INTO TBLDD GOES ON THE STACK
4473 72400 CALL QINLIN ;SEE WHAT HE HAS TO SAY
4474 72440 CHRGET ;SEE WHAT THE FIRST CHARACTER OF INPUT WAS
4475 72480 POP H ;POP OFF POINTER INTO TBLDD
4476 72480 CPI "Y" ;POP OFF CANDIDATE FOR START OF
4477 72480 CPI "N" ;FREE MEMORY
4478 72480 FINFUN: POP D ;POP OFF CANDIDATE FOR START OF
4479 72480 CPI "Y" ;FREE MEMORY
4480 72500 FINFUN: POP D ;FREE MEMORY
4481 72520

```

```

    4482          72540   JZ      HAVFNS   ;HE WANTS IT SO WE ARE DONE
    4483          72560   CPI     "N"
    4484          72580   JNZ     ASKAGN  ;IF A BAD ANSWER
    4485          72600   PUSHM
    4486          72620   ;PUSH ON CALL
    4487          72640   XML
    4488          72660   ;LOCATION THAT WE FIX UP
    4489          72680   LXI    D,ILLFUN ;POINTER INTO TBLO GOES ON THE STACK
    4490          72700   MOV    M,E
    4491          72720   INX    M
    4492          72740   MOV    M,D
    4493          72760   POP    M
    4494          72780   JMP    LUPASK> ;GET TBLO POINTER
    4495          72800   IFN    EXITFNC,<
    4496          72820   ASKAGN1 LXI    M,FNS
    4497          72840   CALL   STROUT ;THE STRING
    4498          003625*  001000  000041
    4499          003625*  000000  004010*
    4500          003625*  001000  003628*
    4501          003627*  001000  000315
    4502          003630*  000000  003523*
    4503          003631*  000000  003625*
    4504          003631*  001000  000315
    4505          003631*  000000  003526*
    4506          003636*  001000  000376
    4507          003636*  000000  000131
    4508          003640*  001000  000621
    4509          003641*  000000  003624*
    4510          003642*  000000  003633*
    4511          003642*  001000  000621
    4512          003643*  000000  003641*
    4513          003643*  000000  003641*
    4514          003646*  001000  000376
    4515          003647*  000000  000101
    4516          003650*  001000  000312
    4517          003651*  000000  003668*
    4518          003652*  000000  003644*
    4519          003652*  001000  000626
    4520          003654*  000000  000016
    4521          003655*  001000  000302
    4522          003656*  000000  003624*
    4523          003657*  000000  003651*
    4524          003660*  001000  000041
    4525          003661*  000000  000000*
    4526          003661*  001000  003636*
    4527          003664*  000000  000121
    4528          003664*  001000  000315*
    4529          003665*  000000  000061*
    4530          003666*  001000  000042
    4531          003667*  000000  000000*
    4532          003667*  000000  003664*
    4533          003671*  001000  000376
    4534          003672*  000000  000101

```

```

    4535          003673*  001000  000312
    4536          003674*  000000  003712*
    4537          003675*  000000  003667*
    4538          003675*  001000  000004
    4539          003676*  000000  000000*
    4540          003700*  000000  000000*
    4541          003711*  001000  000042
    4542          003702*  000000  000002*
    4543          003703*  000000  003677*
    4544          003704*  001000  000004
    4545          003705*  000000  000000*
    4546          003705*  000000  003702*
    4547          003707*  001000  000621
    4548          003708*  000000  002762*
    4549          003711*  000000  003705*
    4550          003712*  000000
    4551          003712*  001000  000053
    4552          003713*  001000  000066
    4553          003714*  000000  000000
    4554          003715*  001000  000043
    4555          003715*  001000  000000
    4556          003717*  000000  000000*
    4557          003720*  000000  003719*
    4558          003721*  001000  000343
    4559          003722*  001000  000021
    4560          003723*  000000  004515*
    4561          003724*  000000  003717*
    4562          003725*  001000  000347
    4563          003725*  001000  000032
    4564          003725*  000000  000000*
    4565          003730*  000000  003723*
    4566          003731*  001000  000321
    4567          003732*  001000  000371
    4568          003733*  001000  000042
    4569          003734*  000000  003413*
    4570          003735*  000000  003727*
    4571          003735*  001000  000353
    4572          003737*  000000  000115
    4573          003740*  000000  000000*
    4574          003741*  002899  003734*
    4575          003742*  001000  000173
    4576          003743*  001000  000225
    4577          003744*  001000  000157
    4578          003745*  001000  000172
    4579          003749*  001000  000234
    4580          003750*  001000  000147
    4581          003751*  001000  000181
    4582          003751*  000000  177769*
    4583          003752*  000000  003748*
    4584          003753*  001000  000011
    4585          003754*  001000  000315
    4586          003755*  000000  003422*
    4587          003756*  000000  003751*
```

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 07([113] 06109 27-AUG-75 PAGE 19-7

F4 MAC 23=AUG=64 06108 SYSTEM INITIALIZATION CODE

4588 003757* 001000 000315 73680 CALL LINPRI IPRINT # OF BYTES FREE
4589 003760* 000000 001766*
4590 003761* 000000 0003755*
4591 003762* 000000 000314*
4592 003763* 000000 004146*
4593 003764* 000000 003762*
4594 003765* 001000 000313
4595 003766* 000000 003630*
4596 003767* 000000 003763*
4597 003770* 001000 000041
4598 003771* 000000 003766*
4599 003772* 000000 000046*
4600 003773* 000000 000042
4601 003774* 000000 000001*
4602 003775* 000000 003713*
4603 003776* 001000 000315
4604 003777* 000000 000000*
4605 004000* 000000 003774*

4606 73800 IFN LPTSW,<
4607 73820 MVI A,4
4608 73840 DCR A,
4609 73860 XRA A
4610 73880 STA PHTFLG##
4611 73900 STA LPTPOS##
4612 004001* 001000 000041
4613 004003* 000000 003777*
4614 004005* 000000 003777*

4615 73940 IFN CUNSSW,<
4616 73960 LXI H,CUNSSD##>
4617 73980 SHLD SCUOE#2

4618 004004* 001000 000042
4619 004005* 000000 000002*
4620 004007* 001000 000351
4621 74000 PCHL

4622 74040 IFE EXTFNC,<
4623 74060 TBLDDI ADR(INITBA)
4624 74080 ADR(FNS1)
4625 74100 ADR(SINPREFIX)
4626 74120 ADR(SIN)
4627 74140 ADR(FNS2)
4628 74160 ADR(RNDFIX)
4629 74180 ADR(RND)
4630 74200 ADR(FNS3)
4631 74220 ADR(SURFIX)
4632 74240 ADR(SQR)
4633 74260 TDASKS:
4634 74280 FNS1 DC"Want SIN"
4635 74300 0
4636 74320 FNS21 DC"Want RND"
4637 74340 0
4638 74360 FNS31 DC"Want SQR"
4639 74380 0>
4640 74400 IFN EXTFNC,<

MAP OF 100-40
INIT READY

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACRO 07([113] 06109 27-AUG-75 PAGE 19-8

F4 MAC 23=AUG=64 06108 SYSTEM INITIALIZATION CODE

4641 004010* 000000 000127 74420 FNS1 DC"Want SIN=CUS=TAN=ATN"
4642 004011* 000000 000101
4643 004013* 000000 000116
4644 004015* 000000 000124
4645 004014* 000000 000040
4646 004015* 000000 000123
4647 004016* 000000 000111
4648 004017* 000000 000116
4649 004020* 000000 000055
4650 004021* 000000 000103
4651 004022* 000000 000117
4652 004023* 000000 000133
4653 004024* 000000 000053
4654 004025* 000000 000124
4655 004026* 000000 000101
4656 004027* 000000 000116
4657 004030* 000000 000055
4658 004031* 000000 000101
4659 004032* 000000 00024
4660 004033* 000000 000116
4661 004034* 000000 000316
4662 004034* 000000 000009
4663 004035* 000000 000013
4664 004035* 000000 000012
4665 004037* 000000 000012
4666 004040* 000000 000012
4667 004041* 000000 000122
4668 004042* 000000 000111
4669 004043* 000000 000124
4670 004044* 000000 000124
4671 004045* 000000 000103
4672 004045* 000000 000116
4673 004047* 000000 000040
4674 004050* 000000 000102
4675 004051* 000000 000131
4676 004052* 000000 000000
4677 004053* 000000 000182
4678 004054* 000000 000111
4679 004055* 000000 000114
4680 004055* 000000 000114
4681 004057* 000000 000040
4682 004060* 000000 000107
4683 004061* 000000 000101
4684 004065* 000000 000124
4685 004065* 000000 000039
4686 004066* 000000 000123
4687 004065* 000000 000040
4688 004065* 000000 000046
4689 004067* 000000 000040
4690 004070* 000000 000120
4691 004071* 000000 000101
4692 004072* 000000 000125
4693 004073* 000000 000114

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACHO 47(113) 06:09 27-AUG-75 PAGE 19-9
F4 MAC 23-AUG-64 06108 SYSTEM INITIALIZATION CODE

4694 004074* 000000 000140
4695 004075* 000000 000101
4696 004076* 000000 000114
4697 004077* 000000 000114
4698 004078* 000000 000105
4699 004131* 000000 000110
4700 004132* 000000 000140
4701 004133* 000000 000048
4702 004134* 000000 000040
4703 004135* 000000 000115
4704 004136* 000000 000117
4705 004137* 000000 000116
4706 004138* 000000 000141
4707 004111* 000000 000105
4708 004112* 000000 000048
4709 004113* 000000 000104
4710 004114* 000000 000101
4711 004115* 000000 000126
4712 004116* 000000 000111
4713 004117* 000000 000104
4714 004118* 000000 000101
4715 004121* 000000 000106
4716 004122* 000000 000106
4717 004123* 000000 000056
4718 004124* 000000 000256
4719 004124* 000000 000015 74520 ACRLF
4720 004125* 000000 000012
4721 004126* 000000 000000 74540 0
4722 004127* 000000 000124
4723 004127* 000000 000124 74560 TTYWID: DC"TERMINAL WIDTH"
4724 004131* 000000 000125
4725 004131* 000000 000122
4726 004132* 000000 000115
4727 004133* 000000 000111
4728 004134* 000000 000116
4729 004135* 000000 000101
4730 004136* 000000 000114
4731 004137* 000000 000040
4732 004140* 000000 000127
4733 004141* 000000 000111
4734 004142* 000000 000124
4735 004143* 000000 000124
4736 004144* 000000 000116
4737 004145* 000000 000316
4738 004145* 000000 000000 74560 0
4739 004146* 000000 000049 74562 WORDS: DC" BYTES FREE"
4740 004147* 000000 000102
4741 004150* 000000 000131
4742 004151* 000000 000124
4743 004152* 000000 000105
4744 004153* 000000 000123
4745 004154* 000000 000040
4746 004155* 000000 000106

MATHPK FOR BASIC MCS 8080 GATES/ALLEN/DAVIDOFF MACHO 47(113) 06:09 27-AUG-75 PAGE 19-10
F4 MAC 23-AUG-64 06108 SYSTEM INITIALIZATION CODE

4747 004156* 000000 000122
4748 004157* 000000 000103
4749 004158* 000000 000103
4750 004159* 000000 000109
4751 004161* 000000 000013 74640 ACRLF
4752 004162* 000000 000012
4753 004163* 000000 000013
4754 004164* 000000 000012
4755 004165* 000000 000101
4756 004165* 000000 000114
4757 004166* 000000 000124
4758 004167* 000000 000101
4759 004171* 000000 000111
4760 004172* 000000 000124
4761 004173* 000000 000040
4762 004174* 000000 000102
4763 004175* 000000 000101
4764 004176* 000000 000123
4765 004177* 000000 000111
4766 004200* 000000 000103
4767 004201* 000000 000040
4768 004202* 000000 000126
4769 004203* 000000 000105
4770 004204* 000000 000122
4771 004205* 000000 000123
4772 004205* 000000 000111
4773 004207* 000000 000117
4774 004207* 000000 000116
4775 004211* 000000 000040
4776 004212* 000000 000063
4777 004213* 000000 000056
4778 004214* 000000 000060
4779 004214* 000000 000268
4780 004215* 000000 000015 74700 ACRLF
4781 004215* 000000 000012
4782 004220* 000000 000185
4783 004221* 000000 000111
4784 004221* 000000 000185
4785 004221* 000000 000111
4786 004222* 000000 000107
4787 004223* 000000 000110
4788 004224* 000000 000124
4789 004225* 000000 000055
4790 004225* 000000 000113
4791 004230* 000000 000040
4792 004230* 000000 000135
4793 004231* 000000 000105
4794 004232* 000000 000122
4795 004233* 000000 000123
4796 004234* 000000 000111
4797 004235* 000000 000117
4798 004236* 000000 000116
4799 004237* 000000 000135

4785 004217* 000000 000133
4786 004220* 000000 000185
4787 004221* 000000 000111
4788 004221* 000000 000107
4789 004223* 000000 000110
4790 004223* 000000 000124
4791 004225* 000000 000055
4792 004230* 000000 000113
4793 004231* 000000 000105
4794 004232* 000000 000122
4795 004233* 000000 000123
4796 004234* 000000 000111
4797 004235* 000000 000117
4798 004236* 000000 000116
4799 004237* 000000 000135

74728 IFE LENGTH,<DC"FOUR-K VERSION">
74740 IFE LENGTH=1,<DC"EIGHT-K VERSION">