

```

4148 005016 000000 005011 59580 MVI A,"?"
4149 005017 001000 000076
4150 005020 000000 000077 59600 OUTCHR
4151 005021 001000 000357 59620 CALL QINLIN
4152 005022 001000 000315
4153 005023 000000 002522
4154 005024 000000 005015
4155
4156
4157
4158 005025 59640 DATBK:
4159 59720 IFE LENGTH,<PUP D
4160 59740 INX H
4161 59760 CALL REDINP
4162 59780 IFN LENGTH,<
4163 59800 IFN STRING,<
4164 59820 LDA VALTYP
4165
4166 005026 001000 000072
4167 005027 000000 001543
4168 005028 000000 005023
4169 005030 001000 000376
4170 005031 000000 000003
4171 005032 001000 000365
4172 005033 001000 000302
4173 005034 000000 005076
4174 005035 000000 005026
4175
4176
4177
4178
4179
4180 005036 001000 000327
4181 005037 001000 000127
4182 005040 001000 000107
4183 005041 001000 000376
4184 005042 000000 000042
4185 005043 001000 000312
4186 005044 000000 000503
4187 005045 000000 005034
4188 005046 001000 000026
4189 005047 000000 000072
4190 005050 001000 000006
4191 005051 000000 000054
4192
4193 005052 001000 000053
4194
4195
4196 005053 001000 000315
4197 005054 000000 007643
4198 005055 000000 005044
4199
4200

```

59640 59660 59680 59700 59720 59740 59760 59780 59800 59820 59840 59860 59880 59900 59920 59940 59960 59980 60000 60020 60040 60060 60080 60100 60120 60140 60160 60180 60200 60220 60240 60260

IFE LENGTH,<PUP D
INX H
CALL REDINP
IFN LENGTH,<
IFN STRING,<
LDA VALTYP
IFE LENGTH=2,<
CPI 3
PUSH PSH
JNZ NUMINS
IFN LENGTH=2,<
ORA A
JZ NUMINS
MOV D,A
MOV B,A
CPI 34
JZ NOWGET
MVI D,""
MVI B,44
DCX H
NOWGET: CALL STRLT2
IFE LENGTH=2,<

DOUBLE PROMPT WHEN WE NEED MORE INPUT
GET A WHOLE LINE AFTER TYPING "?"
IF THE DATA NOW STARTS AT THE BEGINNING
OF THE BUFFER
AND QINLIN LEAVES (H,L)=BUF
POFF OFF POINTER TO THE VARIABLE
FUDGE CHARACTER POINTER
READ A VALUE USING "LET" CODE
IF IT IS NUMERIC OR STRING
IS IT A STRING?
SAVE THE TYPE INFORMATION
IF NUMERIC, USE FIN TO GET IT
IFN LENGTH=2,<
ORA A
JZ NUMINS
INPUT A NUMBER IF NUMERIC
ONLY THE VARIABLE TYPE IS
CHECKED SO AN UNQUOTED STRING
CAN BE ALL DIGITS
CHRGCT
D,A
B,A
34
JZ NOWGET
TERMINATORS OK
MVI D,""
UNQUOTED STRING TERMINATORS
ARE COLON AND COMMA
BACKUP SINCE START CHARACTER MUST BE INCLUDED
IFN THE QUOTED STRING CASE WE DON'T WANT TO
INCLUDE THE STARTING OR ENDING QUOTE
MAKE A STRING DESCRIPTOR FOR THE VALUE
AND COPY IF NECESSARY

```

4201 005056 001000 000361 60280 DNASIG: POP PSH
4202 005057 001000 000353 60320 XCHG
4203 005060 001000 000041 60340 LXI H,STRDN2
4204 005061 000000 005077
4205 005062 000000 005054
4206 005063 001000 000343 60360 XTML
4207 005064 001000 000325 60380 PUSH D
4208 005065 001000 000303 60400 JMP INPCON
4209 005066 000000 004157
4210 005067 000000 005061
4211 005070 001000 000327 60420 NUMINS: CHRGCT
4212 005071 001000 000315 60440 CALL FIN
4213 005072 000000 000000
4214 005073 000000 005066
4215
4216 005074 001000 000303 60460 IFE LENGTH=2,<
4217 005075 000000 005056 60480 JMP DNASIG
4218 005076 000000 005072
4219
4220
4221
4222
4223
4224 005077 60500
4225 60520 IFN LENGTH=2,<
4226 005077 001000 000053 60540 XTML
4227 005100 001000 000327 60560 CALL MOVHF
4228 005101 001000 000312 60580 POP H
4229 005102 000000 005111 60600 STRDN2:
4230 005103 000000 005076 60620 IFN LENGTH,<
4231 005104 001000 000376 60640 DCX H
4232 005105 000000 000054 60660 CHRGCT
4233 005106 001000 000302 60680 JZ TRMOK
4234 005107 000000 004667
4235 005110 000000 005106
4236 005111 001000 000343 60740 TRMOK: XTML
4237 005112 001000 000053 60760 DCX H
4238 005113 001000 000327 60780 CHRGCT
4239 005114 001000 000302 60800 JNZ LUPDT2
4240 005115 000000 004717
4241 005116 000000 005107
4242
4243
4244
4245 005117 001000 000321 60880 POP D
4246 005120 001000 000072 60900 LDA FLGINP
4247 005121 000000 001602
4248 005122 000000 005115
4249 005123 001000 000267 60920 ORA A
4250
4251 60940 IFE LENGTH,<RZ>
4252 005124 001000 000353 60960 XCHG
4253 005125 001000 000302 60980 JNZ RESFIN
61000

```

60280 60320 60340 60360 60380 60400 60420 60440 60460 60480 60500 60520 60540 60560 60580 60600 60620 60640 60660 60680 60700 60720 60740 60760 60780 60800 60820 60840 60880 60900 60920 60940 60960 60980 61000

DNASIG: POP PSH
XCHG
LXI H,STRDN2
XTML
PUSH D
JMP INPCON
NUMINS: CHRGCT
CALL FIN
IFE LENGTH=2,<
JMP DNASIG
IFN LENGTH=2,<
XTML
CALL MOVHF
POP H
STRDN2:
IFN LENGTH,<
DCX H
CHRGCT
JZ TRMOK
TRMOK: XTML
DCX H
CHRGCT
JNZ LUPDT2
OR A
IFE LENGTH,<RZ>
XCHG
JNZ RESFIN

POFF OFF THE TYPE INFORMATION
(D,E)=TEXT POINTER
RETURN LOC
IF (H,L)=PLACE TO STORE VARIABLE VALUE
TEXT POINTER GOES ON
TOO ASSIGNMENT
ASSIGNMENT IS COMPLICATED
EVEN FOR NUMERICS SO USE THE "LET" CODE
IF (H,L) GET POINTER AT VARIABLE
TO THE ASSIGNMENT
GET BACK THE TEXT POINTER
JENDED PROPERLY?
LOOK AT TERMINATOR
AND SET UP CONDITION CODES
NOT ENDING, CHECK FOR COMMA
AND GET ANOTHER VARIABLE
TO FILL WITH DATA
POFF OFF THE POINTER INTO DATA
FETCH THE STATEMENT TYPE FLAG
DON'T UPDATE DATPTR IF IT WAS AN
INPUT STATEMENT
UPDATE DATPTR

```

4254 005126* 000000 003453*
4255 005127* 000000 005121*
4256
4257 005130* 001000 000266 01020 IFN LENGTH,<
4258 01040 ORA H ;COULD HAVE ENDED WITH COMMA OR
4259 005131* 001000 000041 01000 LXI H,EXIGNT ;COLON, BUT SHOULD BE A ZERO
4260 005132* 000000 005142* ;TEXT FOR "EXTRA"
4261 005133* 000000 005120*
4262 005134* 001000 000325 01100 PUSH D ;SAVE THE TEXT POINTER
4263 005135* 001000 000304 01120 CNZ STROUT ;IF WASN'T! REAL END SAY SOMETHING
4264 005136* 000000 007743*
4265 005137* 000000 005132*
4266 005140* 001000 000341 01140 POP H ;GET BACK THE TEXT POINTER
4267 005141* 001000 000511 01160 RET
4268 005142* 000000 000077 01180 EXIGNT: DC"2EXTRA IGNORED"
4269 005143* 000000 000105
4270 005144* 000000 000130
4271 005145* 000000 000124
4272 005146* 000000 000122
4273 005147* 000000 000101
4274 005150* 000000 000040
4275 005151* 000000 000111
4276 005152* 000000 000107
4277 005153* 000000 000116
4278 005154* 000000 000117
4279 005155* 000000 000122
4280 005156* 000000 000105
4281 005157* 000000 000104
4282 005157* 000000 000304
4283 005160* 000000 000015 01200 ACHLF
4284 005161* 000000 000012
4285 005162* 000000 000000 01220 0>
4286 01240 ;
4287 ; SUBROUTINE TO FIND DATA
4288 01260 ; IN THE 4K "DATA" MUST BE AT THE START OF THE LINE
4289 01300 ; SO THE SEARCH IS MADE USING THE LINKS AT THE START OF EACH LINE.
4290 01320 ;
4291 01340 ; IN THE 8K AND EXTENDED THE SEARCH IS MADE BY USING THE EXECUTION CODE
4292 01360 ; FOR DATA TO SKIP OVER STATEMENTS. THE START WORD OF EACH STATEMENT
4293 01380 ; IS COMPARED WITH DATATK, EACH NEW LINE NUMBER
4294 01400 ; IS STORED IN DATLIN SO THAT IF AN ERROR OCCURS WHILE READING
4295 01420 ; DATA THE ERROR MESSAGE WILL GIVE THE LINE NUMBER OF THE
4296 01440 ; ILL-FORMATTED DATA
4297 01460 ;
4298 005163* 001000 000315 01480 DATLOP: IFN LENGTH,<CALL DATA>
4299 005164* 000000 000072*
4300 005165* 000000 005136*
4301
4302 005166* 001000 000267 01500 IFN LENGTH,<POP H>
4303 005166* 001000 000267 01520 DATFND: IFN LENGTH,<
4304 005167* 001000 000302 01540 ORA A
4305 005170* 000000 005214* 01560 JNZ NOWLIN>
4306 005171* 000000 005164*

```

```

4307
4308 005172* 001000 000043 01580 IFN LENGTH,<
4309 005173* 001000 000367 01600 INX H>
4310 005174* 001000 000171 01620 PUSHM A,C ;SAVE POINTER TO THE NEXT STATEMENT
4311 005175* 001000 000260 01640 ORA B ;SEE IF WE ARE AT THE END
4312 005176* 001000 000336 01660 MVI E,ERRROD ;NO DATA IS ERROR ERRROD
4313 005177* 000000 000004
4314 005200* 001000 000312 01700 JZ ERROR ;IF SO COMPLAIN
4315 005201* 000000 002102*
4316 005202* 000000 005170*
4317
4318 005203* 001000 000301 01720 IFN LENGTH,<INX H> ;SKIP PAST LINE #
4319 005204* 001000 000136 01740 IFN LENGTH,<POP B>
4320 005205* 001000 000043 01760 MOV E,H ;GET DATA LINE #
4321 005206* 001000 000126 01800 INX H
4322 005207* 001000 000353 01820 MOV D,M
4323 005210* 001000 000042 01840 XCHG
4324 005211* 000000 001577* SMLD DATLIN
4325 005212* 000000 005201*
4326 005213* 001000 000353 01860 XCHG> ;RESTORE TEXT POINTER
4327 005214* 001000 000327 01880 NOWLIN: CHRGET ;GET THE STATEMENT TYPE
4328 005215* 001000 000376 01900 CPI DATATK ;IS IS "DATA"?
4329 005216* 000000 000203
4330 005217* 001000 000302 01920 JNZ DATLOP ;NOT DATA SO LOOK SOME MORE
4331 005220* 000000 005163*
4332 005221* 000000 005211*
4333
4334 005222* 001000 000303 01940 IFN LENGTH,<POP B>
4335 005223* 000000 005025* JMP DATBK ;CONTINUE READING
4336 005224* 000000 005220*
4337
4338
4339
4340

```

```

4341          02060 SUBTTL NEXT CODE
4342          02000 ;
4343          02100 ; NOTE:
4344          02120 ;
4345          02140 ;
4346          02160 ; A FOR ENTRY ON THE STACK HAS THE FOLLOWING FORMAT:
4347          02180 ;
4348          02200 ;
4349          02220 ; LOW ADDRESS
4350          02240 ; TOKEN (FORK IN HIGH BYTE) 1 BYTES
4351          02260 ; A POINTER TO THE LOOP VARIABLE 2 BYTES
4352          02280 ; A BYTE REFLECTING THE SIGN OF THE INCREMENT 1 BYTE
4353          02300 ; THE STEP 4 BYTES
4354          02320 ; THE UPPER VALUE 4 BYTES
4355          02340 ; THE LINE # OF THE "FOR" STATEMENT 2 BYTES
4356          02360 ; A TEXT POINTER INTO THE "FOR" STATEMENT 2 BYTES
4357          02380 ; HIGH ADDRESS
4358          02400 ;
4359          02420 ; TOTAL 16 BYTES
4360          02440 ;
4361          005225* 001000 000021 02460 NEXT:
4362          005225* 000000 000000 02480 IFN LENGTH,<LXI 0,>CODE>:FOR THE "NEXT"
4363          005225* 000000 005225*
4364          005227* 000000 005225*
4365          02500 ;STATEMENT WITHOUT ANY ARGS
4366          02520 ;THE CALL FNDFOR WITH [0,E]#0
4367          005230* 02540 NEXT:
4368          02560 IFE LENGTH,<
4369          02580 CALL PTRGET> ;MUST HAVE A VARIABLE
4370          02600 IFN LENGTH,< PTRGET> ;GET A POINTER TO THE
4371          005230* 001000 000304
4372          005231* 000000 006505*
4373          005232* 000000 005226*
4374          02640 ;LOOP VARIABLE INTO [0,E]
4375          005233* 001000 000042 02660 SHLD TEMP ;PUT THE TEXT POINTER
4376          005234* 000000 001003*
4377          005235* 000000 005231*
4378          02680 ;IN A TEMP LOCATION
4379          02700 ;IN CASE THE LOOP TERMINATES
4380          005236* 001000 000315 02720 CALL FNDFOR ;TRY TO FIND A FOR ENTRY
4381          005237* 000000 001744*
4382          005240* 000000 005234*
4383          02740 ;ON THE STACK WHOSE VARIABLE NAME
4384          02760 IFN LENGTH,< MATCHES THIS ONES
4385          02800 JNZ NPERR> ;"NEXT" WITHOUT FOR"
4386          005241* 001000 000302
4387          005242* 000000 002100*
4388          005243* 000000 005237*
4389          005244* 001000 000371 02820 SPHL ;SETUP STACK POINTER BY CHOPPING
4390          02840 ;AT THIS POINT
4391          005245* 001000 000325 02860 PUSH D ;PUT THE VARIABLE PTR BACK ON
4392          005246* 001000 000176 02880 MOV A,M ;STEP ONTO THE STACK
4393          005247* 001000 000043 02900 INX H

```

```

4394          005250* 001000 000365 02920 PUSH PSH
4395          005251* 001000 000325 02940 PUSH D ;PUT THE POINTER TO THE LOOP
4396          02960 ;VARIABLE ONTO THE STACK
4397          02980 IFE LENGTH,<
4398          03000 MVI E,ERRNF
4399          03020 JNZ ERROR>
4400          005252* 001000 000315 03040 CALL MOVFM ;STEP VALUE INTO THE FAC
4401          005253* 000000 000000*
4402          005254* 000000 005242*
4403          005255* 001000 000343
4404          03060 ;
4405          005256* 001000 000345 03100 XTHL ;PUT THE POINTER INTO THE
4406          03120 ;FOR ENTRY ONTO THE STACK
4407          005257* 001000 000315 03140 PUSH H ;PUT THE POINTER TO THE LOOP
4408          005260* 000000 000000* ;VARIABLE BACK ONTO THE STACK
4409          005261* 000000 005253* ;ADD THE STEP AND LOOP VARIABLE
4410          005262* 001000 000341 03160 CALL FADDS
4411          03180 ;
4412          005263* 001000 000315 03200 POP H ;POD OFF THE POINTER TO
4413          005264* 000000 000000* ;THE LOOP VARIABLE
4414          005265* 000000 005260* ;MOV FAC INTO LOOP VARIABLE
4415          005266* 001000 000341 03220 POP H
4416          005267* 001000 000315 03240 CALL MOVFM ;GET THE ENTRY POINTER
4417          005270* 000000 000000* ;GET THE FINAL INTO THE REGISTERS
4418          005271* 000000 005266*
4419          005272* 001000 000345 03260 PUSH H ;SAVE THE ENTRY POINTER
4420          005273* 001000 000315 03280 CALL FCOMP ;COMPARE THE NUMBERS
4421          005274* 000000 000714*
4422          005275* 000000 005270*
4423          005276* 001000 000341 03300 POP H ;STILL POINTING TO THE FINAL VALUE
4424          005277* 001000 000301 03320 POP B ;GET THE SIGN OF THE INCREMENT
4425          005300* 001000 000020 03340 SUB B ;SUBTRACT THE INCREMENTS SIGN FROM THAT
4426          03360 ;OF (CURRENT VALUE-FINAL VALUE)
4427          005301* 001000 000315 03380 CALL MOVFM ;GET LINE # OF "FOR" INTO [0,E]
4428          005302* 000000 005270*
4429          005303* 000000 005274*
4430          03400 ;
4431          005304* 001000 000312 03420 JZ LOOPDN ;GET TEXT POINTER OF "FOR" INTO [0,C]
4432          005305* 000000 005320* ;IF SIGN(FINAL-CURRENT)+SIGN(STEP)#0
4433          005306* 000000 005302*
4434          03440 ;
4435          005307* 001000 000353 03460 XCHG ;THEN THE LOOP IS FINISHED
4436          005310* 001000 000042 03480 SHLD CURLIN ;STORE THE LINE #
4437          005311* 000000 001007*
4438          005312* 000000 005305*
4439          005313* 001000 000151 03500 MOV L,C ;SETUP THE TEXT POINTER
4440          005314* 001000 000140 03520 MOV H,B
4441          005315* 001000 000303 03540 JMP NXTCON
4442          005316* 000000 005276*
4443          005317* 000000 005311*
4444          03580 LOOPDN: SPHL ;ELIMINATE THE FOR ENTRY
4445          005320* 001000 000371 03600 ;SINCE [H,L] MOVED ALL
4446          03600

```

BASIC MCS 0000 GATES/ALLEN/DAVIDOFF
F3 MAC 6-SEP-64 05111

MACRO 47(113) 03:12 10-SEP-75 PAGE 19-2
NEXT CODE

```
4447      63620
4448 005321' 001000 000052      63640      LMLD      TEMP      ;THE WAY DOWN THE ENTRY
4449 005322' 000000 001603'      ;RESTORE THE TEXT POINTER
4450 005323' 000000 005310'
4451      63660 IFE      LENGTH,<JMP      NEWSTT>
4452      63680 IFN      LENGTH,<
4453 005324' 001000 000176      63700      MOV      A,M      ;IS THERE A COMMA AT THE END
4454 005325' 001000 000376      63720      CPI      44      ;IF SO LOOK AT ANOTHER
4455 005326' 000000 000054      ;VARIABLE NAME TO "NEXT"
4456 005327' 001000 000302      63740      JNZ      NEWSTT
4457 005330' 000000 003302'
4458 005331' 000000 005322'
4459 005332' 001000 000327      63760      CHRGCT      ;READ FIRST CHARACTER
4460 005333' 001000 000315      63780      CALL      NEXTC>      ;DO NEXT, BUT DON'T ALLOW
4461 005334' 000000 005230'
4462 005335' 000000 005330'
4463      63800
4464      63820
4465      63840      ;BLANK VARIABLE NAME [D,E]=STK PTR
4466      ;AND WILL NEVER MATCH ANY VARPTR
4467      ;USE CALL TO PUT DUMMY "NEWSTT" ENTRY ON
38800 PAGE
```

BASIC MCS 0000 GATES/ALLEN/DAVIDOFF
F3 MAC 6-SEP-64 05111

MACRO 47(113) 03:12 10-SEP-75 PAGE 20
NEXT CODE

4468
4469