

```

6876 010266 010100 000106 34500 MOV B,M ;(B)=HIGH BYTE OF DATA POINTER
6877 010263 010100 000053 34520 ODX H
6878 010264 010100 000116 34540 MOV C,M ;(B,C)=POINTER AT STRING DATA
6879 010263 010100 000345 34550 PUSH H ;SAVE THIS LOCATION SO THE POINTER
6880 ;CAN BE UPDATED AFTER THE STRING IS
6881 ;MOVED
6882 010266 010100 000053 34560 IFN ODX H
6883 ;LENGTH=2,H
6884 34580 ODX H
6885 010267 010100 000156 34620 MOV L,M ;(L)=STRING LENGTH
6886 010270 010100 000046 34640 MVI H,0 ;(M,L) GET CHARACTER COUNT
6887 010271 010100 000000
6888 010272 010100 000011 34660 DAD B ;(M,L)=POINTER BEYOND STRING
6889 010273 010100 000120 34680 MOV D,B
6890 010274 010100 000131 34700 MOV E,C ;(D,E)=ORIGINAL POINTER
6891 010275 010100 000053 34720 ODX H ;DON'T MOVE ONE BEYOND STRING
6892 010276 010100 000104 34740 MOV B,M ;GET TOP OF STRING IN (B,C)
6893 010277 010100 000115 34760 MOV C,L
6894 010300 010100 000052 34780 LHLD FRETOP ;GET TOP OF FREE SPACE
6895 010301 010100 001573
6896 010302 010100 010277
6897 010303 010100 000315 34800 CALL BLTUC ;MOVE STRING
6898 010304 010100 002010
6899 010305 010100 010301
6900 010306 010100 000341 34820 POP M ;GET BACK POINTER TO DESC.
6901 010307 010100 000161 34840 MOV M,C ;SAVE FIXED ADDR
6902 010310 010100 000043 34860 INX M ;MOVE POINTER
6903 010311 010100 000160 34880 MOV M,B ;HIGH PART
6904 010312 010100 000151 34900 MOV L,C
6905 010313 010100 000160 34920 MOV H,B
6906 010314 010100 000053 34940 ODX H ;FIX UP FRETOP
6907 010315 010100 000303 34960 JMP FNOVAR ;AND TRY TO FIND HIGH AGAIN
6908 010316 010100 010045
6909 010317 010100 010304
6910
6911 35000 ;
6912 35020 ; THE FOLLOWING ROUTINE CONCATENATES TWO STRINGS
6913 35040 ; THE FACLO CONTAINS THE FIRST ONE AT THIS POINT,
6914 35060 ; (M,L) POINTS BEYOND THE + SIGN AFTER IT
6915 35080 ;
6916 010320 010100 000305 35100 CAT: PUSH B ;PUT OLD PRECEDENCE BACK ON
6917 010321 010100 000345 35120 PUSH H ;SAVE TEXT POINTER
6918 010322 010100 000052 35140 LHLD FACLO ;GET POINTER TO STRING DESC.
6919 010323 010100 001637
6920 010324 010100 010316
6921 010325 010100 000343 35160 XTHL ;SAVE ON STACK & GET TEXT POINTER BACK
6922 010326 010100 000315 35180 CALL EVAL ;EVALUATE REST OF FORMULA
6923 010327 010100 000041
6924 010330 010100 010325
6925 010331 010100 000343 35200 XTHL ;SAVE TEXT POINTER, GET BACK DESC.
6926 010332 010100 000315 35220 CALL CHKSTR
6927 010333 010100 006230
6928 010334 010100 010327

```

```

6929 010335 010100 000176 35240 MOV A,M
6930 010336 010100 000345 35260 PUSH M
6931 010337 010100 000052 35280 LHLD FACLO ;SAVE DESC, POINTER,
6932 010340 010100 001637 ;GET POINTER TO 2ND DESC.
6933 010341 010100 010335
6934 010342 010100 000345 35300 PUSH M ;SAVE IT
6935 010343 010100 000006 35320 ADD M ;ADD TWO LENGTHS TOGETHER
6936 010344 010100 000036 35340 MVI E,ERRLS ;SEE IF RESULT, LT, 256
6937 010345 010100 000017
6938 010346 010100 000332 35360 JC ERROR ;ERROR "LONG STRING"
6939 010347 010100 002102
6940 010350 010100 010340
6941 010351 010100 000315 35380 CALL STRINI ;GET INITIAL STRING
6942 010352 010100 007024
6943 010353 010100 010347
6944 010354 010100 000321 35400 POP D ;GET 2ND DESC.
6945 010355 010100 000315 35420 CALL FRETMP
6946 010356 010100 010440
6947 010357 010100 010352
6948 010360 010100 000343 35440 XTHL ;SAVE POINTER TO IT
6949 010361 010100 000315 35460 CALL FRETMP2 ;FREE UP 1ST TEMP
6950 010362 010100 010417
6951 010363 010100 010356
6952 010364 010100 000345 35480 PUSH H ;SAVE DESC, POINTER (FIRST)
6953 010365 010100 000052 35500 LHLD DSCTMP+2 ;GET POINTER TO FIRST
6954 010366 010100 001372
6955 010367 010100 010366
6956 010370 010100 000353 35520 XCHG ;IN (D,E)
6957 010371 010100 000315 35540 CALL MOVINS ;MOVE IN THE FIRST STRING
6958 010372 010100 010407
6959 010373 010100 010366
6960 010374 010100 000315 35560 CALL MOVINS ;AND THE SECOND
6961 010375 010100 010407
6962 010376 010100 010372
6963 010377 010100 000041 35580 LXI M,TSTOP ;CAT REENTERS FORMULA EVALUATION AT TSTOP
6964 010400 010100 005357
6965 010401 010100 010375
6966 010402 010100 000343 35600 XTHL
6967 010403 010100 000345 35620 PUSH H
6968 010404 010100 000303 35640 JMP PUTHEW ;TEXT POINTER OFF FIRST
6969 010405 010100 007705 ;THEN RETURN ADDRESS OF TSTOP
6970 010406 010100 010400
6971
6972
6973 010407 010100 000341 35700 MOVINS: POP M ;GET RETURN ADDR
6974 010410 010100 000343 35720 XTHL ;PUT BACK, BUT GET DESC.
6975 35740 IFN LENGTH=2,H
6976 010411 010100 000176 35760 MOV A,M ;(A)=STRING LENGTH
6977 010412 010100 000043 35780 INX M
6978 010413 010100 000116 35800 MOV C,M ;(B,C)=POINTER AT STRING DATA
6979 010414 010100 000043 35820 INX H
6980 010415 010100 000106 35840 MOV B,M
6981 010416 010100 000157 35860 MOV L,A ;(L)=STRING LENGTH

```

```

6982 35980 IFN LENGTH=2,<
6983 35900 PUSHM
6984 35920 PUSHM ;GET LENGTH ON STACK
6985 35940 POP B ;AND POINTER
6986 35960 POP M> ;TEXT POINTER HERE
6987 35980 MOVSTRI INR L ;CHARACTER COUNT HERE
6988 36000 MOVLP: ODR L ;SET CC'S
6989 36020 RZ ;NO BYTE TO MOVE
6990 36040 LDAX B ;GET CHAR
6991 36060 STAX D ;SAVE IT
6992 36080 INX B ;MOVE POINTERS
6993 36100 INX D
6994 36120 JMP MOVLP ;KEEP DOING IT
6995 36140
6996 36160 ;
6997 36180 ; PRETHP IS PASSED A POINTER TO A STRING DESCRIPTOR IN [D,E]
6998 36200 ; THIS VALUE IS RETURNED IN [H,L], ALL THE OTHER REGISTERS ARE MODIFIED.
6999 36220 ; A CHECK TO IS MADE TO SEE IF THE STRING DESCRIPTOR [D,E] POINTS
7000 36240 ; TO IS THE LAST TEMPORARY DESCRIPTOR ALLOCATED BY PUTNEW.
7001 36260 ; IF SO, THE TEMPORARY IS FREED UP BY THE UPDATING OF TEMPPT.
7002 36280 ; IF A TEMPORARY IS FREED UP, A FURTHER CHECK IS MADE TO SEE IF THE
7003 36300 ; STRING DATA THAT THAT STRING TEMPORARY POINTED TO IS THE
7004 36320 ; THE LOWEST PART OF STRING SPACE IN USE.
7005 36340 ; IF SO, PRETHP IS UPDATED TO REFLECT THE FACT THAT THAT SPACE IS NO
7006 36360 ; LONGER IN USE. THIS CAUSES DIFFICULTY FOR ASSIGNMENT ("LET") BECAUSE
7007 36380 ; THOUGH A TEMPORARY IS BEING FREED UP, NAMELY THE VALUE TO THE RIGHT
7008 36400 ; OF THE EQUAL SIGN IN THE "LET", THE ACTUAL DATA
7009 36420 ; IS STILL ACTIVE DATA SINCE A VARIABLE IS BEING SET UP TO POINT
7010 36440 ; AT IT. "LET" POOLS PRETHP BY SETTING THE LENGTH OF THE
7011 36460 ; TEMPORARY TO ZERO TEMPORARILY.
7012 36480 ;
7013 36500 ;
7014 36520 FRESTRI: CALL CHKSTR ;MAKE SURE ITS A STRING
7015 36540
7016 36560
7017 36580 FREFACI: LHLD FACLO
7018 36600
7019 36620
7020 36640 FRETMP: XCHG
7021 36660 FRETMP: LHLD TEMPPT ;FREE UP THE TEMP IN THE FACLO
7022 36680 ;GET TEMP POINTER
7023 36700
7024 36720
7025 36740 DCX H
7026 36760 MOV B,M ;LOOK AT WHAT IS IN THE LAST TEMP
7027 36780 DCX M ;[B,C]=POINTER AT STRING
7028 36800 DCX M ;DECREMENT TEMPPT BY STKSIZ
7029 36820 MOV C,M
7030 36840 DCX H
7031 36860 IFN LENGTH=2,<
7032 36880 DCX M>
7033 36900 COMPAR
7034 36920 XCHG ;SEE IF [D,E] POINT AT THE LAST
7035 36940 RNZ ;RETURN WITH [H,L]
7036 36960 ;POINTING AT CURRENT DESCRIPTOR
7037 36980 ;RETURN NOW IF NOW FREEDING DONE

```

```

7038 36780 SHLD TEMPPT ;UPDATE THE TEMP POINTER SINCE
7039 36800
7040 36820
7041 36840 PUSH D
7042 36860 MOV D,B ;ITS BEEN DECREMENTED BY 4
7043 36880 MOV E,C ;SAVE [D,E] TO RETURN IN [H,L]
7044 36900 DCX D ;[D,E]=POINTER AT STRING
7045 36920 MOV C,M
7046 36940 LHLD FRETPT ;SUBTRACT ONE
7047 36960 ;[C]=LENGTH OF THE STRING FREED UP
7048 36980 ;SEE IF ITS THE FIRST
7049 37000
7050 37020
7051 37040 COMPAR
7052 37060 JNZ NOTLST ;ONE IN STRING SPACE
7053 37080
7054 37100
7055 37120 MOV B,A ;NO SO DON'T ADD
7056 37140 DAD B
7057 37160 SHLD FRETPT ;MAKE [B]=0
7058 37180
7059 37200
7060 37220 MOV B,A
7061 37240 DAD B
7062 37260 SHLD FRETPT ;ADD
7063 37280
7064 37300
7065 37320 NOTLST: POP H
7066 37340 RET ;GET POINTER AT CURRENT DESCRIPTOR
7067 37360
7068 37380 ;
7069 37400 ; THE FUNCTION LEN(S) RETURNS THE LENGTH OF THE
7070 37420 ; STRING PASSED AS AN ARGUMENT
7071 37440 ;
7072 37460 ;
7073 37480 LEN: LXI B,SNGLFT ;CALL SNGLFT WHEN DONE
7074 37500
7075 37520
7076 37540 PUSH B
7077 37560 CALL FRESTR ;LIKE SO
7078 37580 ;FREE UP TEMP POINTED TO BY FACLO
7079 37600
7080 37620
7081 37640 XRA A
7082 37660 MOV D,A ;FORCE NUMERIC FLAG
7083 37680 IFN LENGTH=2,<
7084 37700 STA VALTYP> ;SET HIGH OF [D,E] TO ZERO FOR VAL
7085 37720 MOV A,M
7086 37740 ORA A
7087 37760 RET ;SET CONDITION CODES ON LENGTH
7088 37780 ;RETURN
7089 37800 ;
7090 37820 ; THE FOLLOWING IS THE ASC(S) FUNCTION, IT RETURNS AN INTEGER
7091 37840 ; WHICH IS THE DECIMAL ASCII EQUIVALENT
7092 37860 ;
7093 37880 ASC: CALL LEN1 ;SET UP ORIGINAL STR
7094 37900
7095 37920
7096 37940 JZ FCERR ;NULL STR, BAD ARG.
7097 37960
7098 37980
7099 38000 INX M ;BUMP POINTER

```

### Formula Evaluation

7141	010574*	000000	007772*						
7142	010575*	000000	010366*						
7143	010576*	001000	000301	30200	POP	B		JGET BACK OFFSET	
7144	010577*	001000	000361	30200	POP	H		JGET BACK DESC POINTER,	
7145	010600*	001000	000345	30300	PUSH	H		JPUT KEEP ON STACK	
7146	010601*	001000	000043	30320	INX	H		JMOVE TO STRING POINTER FIELD	
7147				30340	IFN	LENGTH2,<			
7148				30360	INX	H			
7149	010602*	001000	000106	30380	MOV	B,H		JGET POINTER LOW	
7150	010603*	001000	000000	30400	INX	H		J	
7151	010604*	001000	000146	30420	MOV	H,H		JPOINTER HIGH	
7152	010605*	001000	000150	30440	MOV	H,B		JGET LOW IN L	
7153	010606*	001000	000000	30460	MVI	B,0		JGET READY TO ADD OFFSET TO POINTER	
7154	010607*	000000	000000						
7155	010608*	000000	000011	30480	DAD	B		JADD IT	
7156	010611*	000000	000104	30500	MOV	H,0		JGET OFFSET POINTER IN (B,C)	
7157	010612*	000000	000115	30520	MOV	C,L			
7158	010613*	001000	000215	30540	CALL	STRADD		JSAVE INFO IN DSCTMP	
7159	010614*	000000	007627*						
7160	010615*	000000	010374*						
7161	010616*	001000	000157	30560	MOV	L,A		JGET# OF CHARS TO MOVE IN L	
7162	010617*	001000	000315	30580	CALL	MUVSTR		JMOVE THEM IN	
7163	010620*	000000	010417*						
7164	010621*	000000	010614*						
7165	010622*	001000	000321	30600	POP	0		JGET BACK DESC. POINTER	
7166	010623*	001000	000315	30620	CALL	FRMPT		JFREE IT UP.	
7167	010624*	000000	010446*						
7168	010625*	000000	010620*						
7169	010626*	000000	000303		JMP	PUTNEW		JPUT TEMP IN TEMP LIST	
7170	010627*	000000	007705*						
7171	010636*	000000	010624*						
7172									
7173	010631*	001000	000315	30600	RIGHTS: CALL	PREAM		JCHECK ARG	
7174	010632*	000000	010764*						
7175	010633*	000000	010627*						
7176	010634*	001000	000321	30700	POP	0		JGET DESC. POINTER	
7177	010635*	001000	000325	30720	PUSH	0		JSAVE BACK FOR LEFT	
7178	010636*	000000	000032	30740	LDAX	0		JGET PRESENT LEN OF STR	
7179	010637*	001000	000226	30760	SUB	B		JSUBTRACT 2ND PAR	
7180	010640*	001000	000325	30780	JMP	LEFT3		JCONTINUE WITH LEFT CODE	
7181	010641*	000000	010556*						
7182	010642*	000000	010632*						
7183				30800	J				
7184				30820	J	MID (3,*) RETURNS STR WITH CHARS FROM # POSITION			
7185				30840	J	ONNARU, IF # IS GT LEN(3) THEN RETURN NULL STRING,			
7186				30860	J	MID (3,*,*) RETURNS STR WITH CHARS FROM # POSITION			
7187				30880	J	FOR 3# CHARS, IF #2 GOES PAST END OF STRING, RETURN			
7188				30900	J	AS NULL AS POSSIBLE.			
7189				30920	J				
7190	010643*	001000	000353	30940	MDS: XCHG			JPUT THE TEXT POINTER IN (H,L)	
7191	010644*	001000	000176	30960	MOV	A,H		JGET THE FIRST CHARACTER	
7192	010645*	001000	000315	30980	CALL	PREAM2		JGET OFFSET OFF STACK AND MAKE	
7193	010646*	000000	010767*						

```

7194 010647* 000000 010641*
7195
7196 010650* 001000 000305 39080
7197 010651* 001000 000306 39020 PUSH B ;SURE DOES NOT = 0.
7198 010652* 000000 000377 39040 MVI E,255 ;PUT OFFSET ON TO THE STACK
7199 010653* 001000 000376 39060 CPI "J" ;IF TWO ARG GUY, TRUNCATE.
7200 010654* 000000 000051
7201 010655* 001000 000312 39080 JZ MID2 ;[E] SAYS USE ALL CHARS
7202 010656* 000000 010665*
7203 010657* 000000 010666*
7204
7205 010660* 001000 000317 39100
7206 010661* 000000 000336 39120 SYNCHK 44 ;IF ONE ARGUMENT THIS IS CORRECT
7207 010662* 001000 000315 39140 CALL GETBYT ;CONMA? MUST DELINEATE 3RD ARG.
7208 010663* 000000 011020* ;GET ARGUMENT IN [E]
7209 010664* 000000 010656*
7210 010665* 001000 000317 39160 MID2: SYNCHK "J" ;MUST BE FOLLOWED BY )
7211 010666* 000000 000051
7212 010667* 001000 000361 39180 POP PSW ;GET OFFSET BACK IN A
7213 010670* 001000 000343 39200 XTHL ;SAVE TEXT POINTER, GET DESC.
7214 010671* 001000 000001 39220 LXI B,LEF12 ;WHERE TO RETURN TO,
7215 010672* 000000 010664*
7216 010673* 000000 010653*
7217 010674* 001000 000305 39240 PUSH B ;GOES ON STACK
7218 010675* 001000 000075 39260 DCR A ;SUB ONE FROM OFFSET
7219 010676* 000000 000276 39280 CHP M ;GET PRESENT LEN OF STR
7220 010677* 001000 000000 39300 MVI B,0 ;ASSUME NULL LENGTH STR
7221 010700* 000000 000000
7222 010701* 001000 000320 39320 RNC ;YES, JUST USE NULL STR
7223 010702* 001000 000017 39340 MOV C,A ;SAVE OFFSET OF CHARACTER POINTER
7224 010703* 001000 000176 39360 MOV A,M ;GET PRESENT LEN OF STR
7225 010704* 001000 000221 39380 SUB C ;SUBTRACT INDEX (2ND ARG)
7226 010705* 001000 000273 39400 CHP E ;IS IT TRUNCATION
7227 010706* 001000 000107 39420 MOV B,A ;GET CALCD LENGTH IN B
7228 010707* 001000 000330 39440 RC ;IF NOT USE PARTIAL STR
7229 010710* 001000 000103 39460 MOV B,E ;USE TRUNCATED LENGTH
7230 010711* 001000 000311 39480 RET> ;RETURN TO LEFT2
7231 39500 IFN LENGTH,<
7232 ;
7233 ; THE FOLLOWING FUNCTIONS ALLOW THE
7234 ; USER FULL ACCESS TO THE ALTAIR I/O PORTS
7235 ; INP[CHANNEL#] RETURNS AN INTEGER WHICH IS THE STATUS
7236 ; OF THE CHANNEL, OUT CHANNEL#,VALUE PUTS OUT THE INTEGER
7237 ; VALUE ON CHANNEL #, IT IS A STATEMENT, NOT A FUNCTION.
7238 ;
7239 010712* 001000 000315 39680 FNINP: CALL CONINT ;GET INTEGER CHANNEL #
7240 010713* 000000 011025*
7241 010714* 000000 010674*
7242 010715* 001000 000062 39680 STA INPHRD+1 ;GEN INP INSTN
7243 010716* 000000 010721*
7244 010717* 000000 010715*
7245 010720* 001000 000333 39700 INPHRD: IN 0 ;THE INP INSTN
7246 010721* 000000 000000

```

```

7247 010722* 001000 000303 39720 JMP SNGFLT ;SNGFLT RESULT
7248 010723* 000000 007400*
7249 010724* 000000 010716*
7250
7251 010725* 001000 000315 39760 FNOUT: CALL SETIO ;GET READY
7252 010726* 000000 011005*
7253 010727* 000000 010725*
7254 010730* 001000 000323 39780 OUTWRD: OUT 0 ;DO IT
7255 010731* 000000 000000
7256 010732* 001000 000311 39800 RET ;AND THATS ALL
7257 ;
7258 39840 ; THE WAIT CHANNEL#,MASK,MASK2 WAITS UNTIL THE STATUS
7259 ; RETURNED BY CHANNEL# IS NON ZERO WHEN XORED WITH MASK2
7260 ; AND THEN ANDED WITH MASK, IF MASK2 IS NOT PRESENT IT IS ASSUMED
7261 ; TO BE ZERO.
7262 ;
7263 010733* 001000 000315 39920 ;
7264 010734* 000000 011003* 39940 FNWAIT: CALL SETIO ;SET UP FOR WAIT
7265 010735* 000000 010726*
7266 010736* 001000 000305 39960 PUSH PSW ;SAVE THE MASK
7267 010737* 001000 000036 39980 MVI E,0 ;DEFAULT MASK2 TO ZERO
7268 010740* 000000 000000
7269 010741* 001000 000053 40000 DCX H
7270 010742* 001000 000327 40020 CHRGRT
7271 010743* 001000 000312 40040 JZ NUTTHR ;SEE IF THE STATEMENT ENDED
7272 010744* 000000 010753* ;IF NO THIRD ARGUMENT SKIP THIS
7273 010745* 000000 010734*
7274 010746* 001000 000317 40060 SYNCHK 44 ;MAKE SURE THERE IS A ","
7275 010747* 000000 000054 40080 CALL GETBYT
7276 010750* 001000 000315
7277 010751* 000000 011026*
7278 010752* 000000 010744*
7279 010753* 001000 000301 40100 NUTTHR: POP B ;RGET THE "AND" MASK
7280 010754* 001000 000333 40120 STAINP: IN 0 ;THE INPUT INSTN
7281 010755* 000000 000000
7282 010756* 001000 000053 40140 XRA E ;XOR WITH MASK2
7283 010757* 001000 000040 40160 ANA B ;AND WITH MASK
7284 010760* 001000 000312 40180 JZ STAINP ;LOOP UNTIL RESULT IS NON-ZERO
7285 010761* 000000 010754*
7286 010762* 000000 010751*
7287
7288 40200 ;NOTE: THIS LOOP CANNOT BE CONTROL-C'ED
7289 40220 ;UNLESS THE WAIT IS BEING DONE ON CHANNEL
7290 40240 ;ZERO, HOWEVER A RESTART AT 0 IS OK.
7291 010763* 001000 000311 40260
7292 IFN STRING,<
7293 010764* 001000 000353 40300 PREAM: XCHG ;USED BY RIGHTS AND LEFTS FOR PARAMETER CHECKING AND SETUP
7294 010765* 001000 000317 40320 SYNCHK "J" ;PUT THE TEXT POINTER IN [H,L]
7295 010766* 000000 000051 40340 ;PARAM LIST SHOULD END
7296
7297 010767* 001000 000301 40360 ;USED BY MID2 FOR PARAMETER CHECKING AND SETUP
7298 010770* 001000 000321 40380 PREAM2: POP B ;GET RETURN ADDR OFF STACK
7299 010771* 001000 000305 40400 POP D ;GET LENGTH OF ARG OFF STACK
7300 40420 PUSH B ;SAVE RETURN ADDR BACK ON

```