

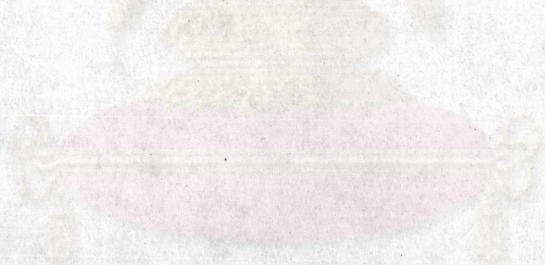
เอกสารอ้างอิง

1. พันเลิศ ประทีปศรี วิทยานิพนธ์เรื่องการศึกษาพัฒนาอินเทอร์พรีเตอร์ภาษาเบสิก
2. PDP-11 BASIC-PLUS-2(V1.6) LANGUAGE REFERENCE MANUAL,
MASSACHUSETTS 01754, DIGITAL EQUIPMENT CORPORATION,
JANUARY 1981
3. CARMA L. MCCLURE, REDUCING COBOL COMPLEXITY THROUGH
STRUCTURED PROGRAMMING NEW YORK, VAN NOSTRAND REINHOLD
COMPANY, 1978
4. BASIC-PLUS-2 RETSIE USER' MASSACHUSETTES 01754,
DIGITAL EQUIPMENT CORPORATION JANUARY 1981
5. NIPPON ELECTRIC CO.,LTD., PC-8001B
REFERENCE MANUAL, 1981
6. WILLIAM PAYNE AND PATRICIA PAYNE, IMPLEMENTING BASICS HOW
BASICS WORK VIRGINIA, A PRENTICE-HALL COMPANY

ศูนย์วิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก



ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.
รายละเอียดโปรแกรมของวิทยาลัยเทคนิค

** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 1

TABLE OF COMMAND STATEMENT

6000 2F43	ORG 6000H	
6002 5941	DW 432FH	;END
6004 084A	DW 4159H	;FOR
6006 BE45	DW 4A08H	;NEXT
6008 DA48	DW 45BEH	;DATA
600A 7D73	DW 48DAH	;INPUT
600C 3949	DW 737DH	;DIM ;XXXXXXXX 4E37H
600E DE45	DW 4939H	;READ
6010 6D45	DW 45DEH	;LET
6012 3D45	DW 456DH	;GOTO
6014 0247	DW 453DH	;RUN
6016 0243	DW 4702H	;IF
6018 8970	DW 4302H	;RESTORE
601A 2C70	DW 7089H	;GOSUB ;XXXXXXXX 4555H
601C C045	DW 702CH	;RETURN ;XXXXXXXX 45A3H
601E 2A43	DW 45C0H	;REM
6020 4247	DW 432AH	;STOP
6022 E844	DW 4742H	;PRINT
6024 0C57	DW 44E8H	;CLEAR
6026 E03D	DW 570CH	;LIST
6028 4246	DW 3DE0H	;NEW
602A B356	DW 4642H	;ON
602C CC50	DW 56B3H	;WAIT
602E 1859	DW 50CCH	;DEF
6030 8343	DW 5918H	;POKE
6032 C01E	DW 4383H	;CONT
6034 101F	DW 1EC0H	;CSAVE
6036 AD56	DW 1F10H	;CLOAD
6038 3A47	DW 56ADH	;OUT
603A 0757	DW 473AH	;LPRINT
603C 8408	DW 5707H	;LLIST
603E 4308	DW 08B4H	;CONSOL
6040 C045	DW 0843H	;WIDTH
6042 9643	DW 45C0H	;ELSE
6044 9743	DW 4396H	;TRON
6046 9C43	DW 4397H	;TROFF
6048 DF43	DW 439CH	;SWAP
604A C446	DW 43DFH	;ERASE
604C 8C46	DW 46C4H	;ERROR
604E D958	DW 468CH	;RESUME
6050 CF46	DW 58D9H	;DELETE
6052 ED5A	DW 46CFH	;AUTO
6054 5B44	DW 5AEDH	;RENUM
6056 5E44	DW 445BH	;DEFSTR
6058 6144	DW 445EH	;DEFINT
605A 6444	DW 4461H	;DEFSGN
605C 7748	DW 4464H	;DEFDBL
605E 0507	DW 4877H	;LINE
6060 B806	DW 0705H	;PRESET
6062 410D	DW 06B8H	;PSET
6064 2FF1	DW 0D41H	;BEEP
6066 4313	DW 0F12FH	;FORMAT
6068 5109	DW 1343H	;KEY
606A 1C6E	DW 0951H	;COLOR
	DW 06E1CH	; ;GOSUBL

** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 2

606C E16F	DW	6FE1H	; ;RETURNL
606E C067	DW	067C0H	; ;LOCAL
6070 3271	DW	07132H	; ;GLOBAL
6072 306D	DW	05D30H	; ;WHEN
6074 516C	DW	06C51H	; ;LOOP
6076 616D	DW	06D61H	; ;ENDLOOP
6078 686A	DW	06A08H	; ;IFS
607A 136C	DW	06C13H	; ;THENDO
607C 286C	DW	06C28H	; ;ELSEDO
607E F46B	DW	06BF4H	; ;ENDDO
6080 3D6C	DW	06C3DH	; ;ENDIF
6082 0D71	DW	0710DH	; ;LABEL
6084 50F1	DW	0F150H	; REMOVE
6086 53F1	DW	0F153H	; MOUNT
6088 FFF0	DW	0FFF0H	; OPEN
608A 02F1	DW	0F102H	; FIELD
608C 8618	DW	01886H	; GET
608E 9118	DW	1891H	; PUT
6090 56F1	DW	0F156H	; SET
6092 35F1	DW	0F135H	; CLOSE
6094 38F1	DW	0F138H	; LOAD
6096 3BF1	DW	0F13BH	; MERGE
6098 4DF1	DW	0F14DH	; FILES
609A 3EF1	DW	0F13EH	; NAME
609C 41F1	DW	0F141H	; KILL
609E 44F1	DW	0F144H	; LSET
60A0 47F1	DW	0F147H	; RSET
60A2 44F1	DW	0F14AH	; SAVE
60A4 26F1	DW	0F126H	; LFILES
60A6 6222	DW	02262H	; INIT
60A8 9207	DW	08792H	; LOCATE

;FUNCTION ADDRESS TABLE I

60AA F954	DW	54F9H	; LEFT
60AC 2955	DW	5529H	; RIGHT
60AE 3255	DW	5532H	; MID
60B0 8626	DW	2686H	; SGN
60B2 3F28	DW	283FH	; INT
60B4 7126	DW	2671H	; ABS
60B6 A131	DW	31A1H	; SQR
60B8 8332	DW	3283H	; RND
60BA FC32	DW	32FCH	; SIN
60BC 0325	DW	2503H	; LOG
60BE F331	DW	31F3H	; EXP
60C0 F632	DW	32F6H	; COS
60C2 5D33	DW	335DH	; TAN
60C4 7233	DW	3372H	; ATN
60C6 5150	DW	5051H	; FRE
60C8 A156	DW	56A1H	; INP
60CA 7950	DW	5079H	; PUS
60CC 8C54	DW	548CH	; LEN
60CE 7A52	DW	527AH	; STR
60D0 5355	DW	5553H	; VAL
60D2 9854	DW	5498H	; ASC
60D4 A853	DW	53A8H	; CHR
60D6 1159	DW	5911H	; PEEK
60D8 DF54	DW	54DFH	; SPACE
60DA 7052	DW	5270H	; OCT
60DC 7552	DW	5275H	; HEX

DUAD-PC ASSEMBLER (A.C.P.)

PAGE 3

60DE 7450	DW	5074H	;LPOS
60E0 F920	DW	20F9H	;PORT
60E2 23F1	DW	0F123H	;DEC
60E4 20F1	DW	0F120H	;BCD
60E6 7F27	DW	0277FH	;CINT
60E8 B327	DW	27B3H	;CSNG
60EA DF27	DW	27DFH	;CDBL
60EC 2C28	DW	202CH	;FIX
60EE E1F0	DW	0F0E1H	;CVI
60F0 E4F0	DW	0F0E4H	;CVS
60F2 E7F0	DW	0F0E7H	;CVD
60F4 59F1	DW	0F159H	;DSKF
60F6 EAF0	DW	0F0EAH	;EOF
60F8 EDF0	DW	0F0EDH	;LOC
60FA F0F0	DW	0F0F0H	;LOF
60FC 68F1	DW	0F168H	;FPOS
60FE F3F0	DW	0F0F3H	;MKI
6100 F4F0	DW	0F0F6H	;MKS
6102 F9F0	DW	0F0F9H	;MKD

KEY WORD TABLE

6104 3061	DW	6130H	;TOPA
6106 4E61	DW	614EH	;TOPB
6108 5761	DW	6157H	;TOPC
610A 9E61	DW	619EH	;TOPD
610C D061	DW	61D8H	;TOPE
610E 1962	DW	6219H	;TOPF
6110 3962	DW	6239H	;TOPG
6112 5762	DW	6257H	;TOPH
6114 5C62	DW	625CH	;TOPI
6116 8362	DW	6283H	;TOPJ
6118 8462	DW	6284H	;TOPK
611A 8C62	DW	628CH	;TOPL
611C DA62	DW	62DAH	;TOPM
611E F862	DW	62FBH	;TOPN
6120 0763	DW	6307H	;TOPO
6122 1863	DW	6318H	;TOPP
6124 3F63	DW	633FH	;TOPQ
6126 4063	DW	6340H	;TOPR
6128 7D63	DW	637DH	;TOPS
612A B863	DW	6388H	;TOPT
612C DA63	DW	63DAH	;TOPU
612E E363	DW	63E3H	;TOPV
6130 ED63	DW	63EDH	;TOPW
6132 F763	DW	63F7H	;TOPX
6134 FB63	DW	63FBH	;TOPY
6136 FC63	DW	63FCH	;TOPZ

6138 4EC4F0	TOPA:	DB	'N', 'D'+00H, 0F0H
613B 42D306		DB	'B', 'S'+00H, 6
613E 54CE0E		DB	'T', 'N'+00H, 0EH
6141 53C215		DB	'S', 'B'+00H, 15H
6144 5554CFA9		DB	'UT', 'D'+00H, 0A9H
6148 545452A4		DB	'TTR', 'S'+00H, 0E7H
614C E7			
614D 00		DB	0
614E 4344A41E	TOPB:	DB	'CD', 'S'+00H, 1EH

XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 4

```

6152 454508B2      DB 'EE', 'P'+80H, 0B2H
6156 00           DB 0

6157 4F4E334F TOPC: DB 'ONSOL', 'E'+80H, 9FH
6158 4CC59F
615E 4C4F53C5      DB 'LOS', 'E'+80H, 0CAH
6162 CA
6163 4F4ED499      DB 'ON', 'T'+80H, 99H
6167 4C4541D2      DB 'LEA', 'R'+80H, 92H
6168 92
616C 4C4F41C4      DB 'LOA', 'D'+80H, 9BH
6170 9B
6171 534156C5      DB 'SAV', 'E'+80H, 9AH
6175 9A
6176 53524C49      DB 'SRLI', 'N'+80H, 0E6H
617A CEE6
617C 494ED41F      DB 'IN', 'T'+80H, 1FH
6180 534EC720      DB 'SN', 'G'+80H, 20H
6184 4442CC21      DB 'DB', 'L'+80H, 21H
6188 56C923        DB 'U', 'I'+80H, 23H
618B 56D324        DB 'U', 'S'+80H, 24H
618E 56C425        DB 'U', 'D'+80H, 25H
6191 4FD30C        DB 'O', 'S'+80H, 0CH
6194 4852A416      DB 'HR', 'S'+80H, 16H
6198 4F4C4FD2      DB 'OLO', 'R'+80H, 0B5H
619C B5
619D 00           DB 0

619E 4154C184 TOPD: DB 'AT', 'A'+80H, 84H
61A2 49CD86        DB 'I', 'M'+80H, 86H
61A5 45465354      DB 'EFST', 'R'+80H, 0ABH
61A9 D2AB
61AB 4546494E      DB 'EFIN', 'T'+80H, 0ACH
61AF D4AC
61B1 4546534E      DB 'EF8N', 'G'+80H, 0ADH
61B5 C7AD
61B7 45464442      DB 'EFDB', 'L'+80H, 0AEH
61BB CCAE
61BD 45C697        DB 'E', 'F'+80H, 97H
61C0 454C4554      DB 'ELET', 'E'+80H, 0A8H
61C4 C5A8
61C6 534849A4      DB 'SKI', 'S'+80H, 0E8H
61CA EB
61CB 5348C626      DB 'SK', 'F'+80H, 26H
61CF 45C31D        DB 'E', 'C'+80H, 1DH
61D2 415445A4      DB 'ATE', 'S'+80H, 0EBH
61D6 EB
61D7 00           DB 0

61D8 4C534544 TOPE: DB 'LSED', 'D'+80H, 0BFH
61DC CFBF
61DE 4E4444CF      DB 'NDD', 'D'+80H, 0C8H
61E2 C8
61E3 4E4449C6      DB 'NDI', 'F'+80H, 0C1H
61E7 C1
61E8 4E444C4F      DB 'NDLOO', 'P'+80H, 0BCH
61EC 4FD08C
61EF 4EC481        DB 'N', 'D'+80H, 81H
61F2 4C53C5A1      DB 'LS', 'E'+80H, 0A1H
61F6 524153C5      DB 'RAS', 'E'+80H, 0A5H
61FA A5

```


XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 5

```

41FB 52524FD2      DB  'RRO', 'R'+80H, 0A6H
41FF A6
4208 52CCDF        DB  'R', 'L'+80H, 0DFH
4209 52D2E0        DB  'R', 'R'+80H, 0E0H
420A 58495420      DB  'XIT WHE', 'N'+80H, 0BAH
420E BA
420F 58D00B        DB  'X', 'P'+80H, 0BH
4212 4FC627        DB  'O', 'F'+80H, 27H
4215 51D6FB        DB  'Q', 'U'+80H, 0FBH
4218 00            DB  0

4219 38524D41 TOPF: DB  'ORMA', 'T'+80H, 0B3H
421D D4B3
421F 4FD282        DB  'O', 'R'+80H, 82H
4222 49454CC4      DB  'IEL', 'D'+80H, 0C6H
4226 C6
4227 494C45D3      DB  'ILE', 'S'+80H, 0CDH
422B CD
422C CEDC          DB  'N'+80H, 0DCH
422E 52C50F        DB  'R', 'E'+80H, 0FH
4231 49DB22        DB  'I', 'X'+80H, 22H
4234 584FD32A      DB  'PO', 'S'+80H, 2AH
4238 00            DB  0

4239 4F54CF89 TOPG: DB  'OT', 'O'+80H, 89H
423D 4F2054CF      DB  'O T', 'O'+80H, 89H
4241 89
4242 4C4F4241      DB  'LOBA', 'L'+80H, 0B9H      ;XXXXXXXX GOBOL
4246 CCB9
4248 4F535542      DB  'OSUB', 'L'+80H, 0B6H      ;XXXXXXXX GOSUBL
424C CCB6
424E 4F5355C2      DB  'OSU', 'B'+80H, 8DH
4252 8D
4253 45D4C7        DB  'E', 'T'+80H, 0C7H
4256 00            DB  0

4257 4558A41A TOPH: DB  'EX', 'S'+80H, 1AH
425B 00            DB  0

425C 4E5855D4 TOPI: DB  'NPU', 'T'+80H, 85H
4260 85
4261 46D3BD        DB  'F', 'S'+80H, 0BDH
4264 C68B          DB  'F'+80H, 8BH
4266 4E5354D2      DB  'NST', 'R'+80H, 0E3H
426A E3
426B 4ED405        DB  'N', 'T'+80H, 5H
426E 4ED810        DB  'N', 'P'+80H, 10H
4271 4DD0FC        DB  'M', 'P'+80H, 0FCH
4274 4E49D4D4      DB  'NI', 'T'+80H, 0D4H
4278 4E4B4559      DB  'NKEY', 'S'+80H, 0E9H
427C A4E9
427E 4545C34C      DB  'EE', 'E'+80H, 6CH
4282 00            DB  0

4283 00            TOPJ: DB  0

4284 494CCCCF TOPK: DB  'IL', 'L'+80H, 0CFH
4288 45D9B4        DB  'E', 'Y'+80H, 0B4H
428B 00            DB  0

```


XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 6

628C 45D48B			
628F 414245CC			
6293 C2			
6294 4F4341CC			
6298 B8			
6299 4F4FD8BB			
629D 4F434154			
62A1 C5D5			
62A3 494EC5AF			
62A7 4F41C4CB			
62AB 5345D4D8			
62AF 5852494E			
62B3 D49D			
62B5 4C4953D4			
62B9 9E			
62BA 584FD31B			
62BE 4953D493			
62C2 46494C45			
62C6 D3D3			
62C8 4FC78A			
62CB 4FC328			
62CE 45CE12			
62D1 454654A4			
62D5 81			
62D6 4FC629			
62D9 88			
62DA 4F554ED4			
62DE C4			
62DF 455247C5			
62E3 CC			
62E4 4FC4FD			
62E7 4B49A42B			
62EB 4B53A42C			
62EF 4B44A42D			
62F3 4944A483			
62F7 88			
62F8 4558D483			
62FC 414DC5CE			
6308 45D794			
6303 4FD4DE			
6306 88			
6307 55D49C			
630A 4E8895			
630D 5845CEC5			
6311 D2F9			
6313 4354A419			
6317 88			
6318 55D4C8			
631B 4F4BC598			
631F 52494ED4			
6323 91			
6324 4FD311			
6327 4545CB17			
632B 4F52D41C			
632F 5345D4B1			
6333 52455345			
TOPL:	DB	'E', 'T'+80H, 88H	
	DB	'ABE', 'L'+80H, 8C2H	
	DB	'OCA', 'L'+80H, 8B8H	
	DB	'OD', 'P'+80H, 8BBH	
	DB	'OCAT', 'E'+80H, 8D5H	
	DB	'IN', 'E'+80H, 8AFH	
	DB	'OA', 'D'+80H, 8CBH	
	DB	'SE', 'T'+80H, 8D8H	
	DB	'PRIN', 'T'+80H, 9DH	
	DB	'LIS', 'T'+80H, 9EH	
	DB	'PO', 'S'+80H, 1BH	
	DB	'IS', 'T'+80H, 93H	
	DB	'FILE', 'S'+80H, 8D3H	
	DB	'O', 'O'+80H, 8AH	
	DB	'O', 'C'+80H, 28H	
	DB	'E', 'N'+80H, 12H	
	DB	'EFT', 'S'+80H, 1	
	DB	'O', 'F'+80H, 29H	
	DB	8	
TOPM:	DB	'OUN', 'T'+80H, 8C4H	
	DB	'ERG', 'E'+80H, 8CCH	
	DB	'O', 'D'+80H, 8FDH	
	DB	'KI', 'S'+80H, 28H	
	DB	'KS', 'S'+80H, 2CH	
	DB	'KD', 'S'+80H, 2DH	
	DB	'ID', 'S'+80H, 3	
	DB	8	
TOPN:	DB	'EX', 'T'+80H, 83H	
	DB	'AM', 'E'+80H, 8CEH	
	DB	'E', 'W'+80H, 94H	
	DB	'O', 'T'+80H, 8DEH	
	DB	8	
TOPD:	DB	'U', 'T'+80H, 9CH	
	DB	'N', '+80H, 95H	
	DB	'PE', 'N'+80H, 8C5H	
	DB	'R'+80H, 8F9H	
	DB	'CT', 'S'+80H, 19H	
	DB	8	
TOPP:	DB	'U', 'T'+80H, 8CBH	
	DB	'OK', 'E'+80H, 98H	
	DB	'RIN', 'T'+80H, 91H	
	DB	'O', 'S'+80H, 11H	
	DB	'EE', 'K'+80H, 17H	
	DB	'OR', 'T'+80H, 1CH	
	DB	'SE', 'T'+80H, 8B1H	
	DB	'RESE', 'T'+80H, 8B8H	

XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 7

```

6337 D480
6339 4F494ED4 DB 'OIN', 'T'+80H, 0EFH
633D EF DB 0
633E 00 DB 0
633F 00 TOPQ: DB 0
6340 4541C487 TOPR: DB 'EA', 'D'+80H, 87H
6344 55CE8A DB 'U', 'N'+80H, 8AH
6347 4553544F DB 'ESTOR', 'E'+80H, 8CH
634B 52C58C
634E 45545552 DB 'ETURN', 'L'+80H, 0B7H
6352 4ECCB7
6355 45545552 DB 'ETUR', 'N'+80H, 8EH
6359 CE8E
635B 45404F56 DB 'EMOV', 'E'+80H, 0C3H
635F C5C3
6361 45CD8F DB 'E', 'M'+80H, 8FH
6364 4553554D DB 'ESUM', 'E'+80H, 0A7H
6368 C5A7
636A 5345D4D1 DB 'SE', 'T'+80H, 0D1H
636E 49474854 DB 'IGHT', 'S'+80H, 2
6372 A402
6374 4EC408 DB 'N', 'D'+80H, 8
6377 454E55CD DB 'ENU', 'M'+80H, 0AAH
637B AA
637C 00 DB 0
637D 544FD090 TOPS: DB 'TO', 'P'+80H, 90H
6381 5741D8A4 DB 'NA', 'P'+80H, 0A4H
6385 45D4C9 DB 'E', 'T'+80H, 0C9H
6388 4156C5D2 DB 'AV', 'E'+80H, 0D2H
638C 5043A8DD DB 'PC', 'C'+80H, 0DDH
6390 5445D8DA DB 'TE', 'P'+80H, 0DAH
6394 47CE04 DB 'G', 'N'+80H, 4
6397 51D207 DB 'Q', 'R'+80H, 7
639A 49CE09 DB 'I', 'N'+80H, 9
639D 5452A413 DB 'TR', 'S'+80H, 13H
63A1 5452494E DB 'TRING', 'S'+80H, 0E1H
63A5 47A4E1
63A8 50414345 DB 'PACE', 'S'+80H, 10H
63AC A418
63AE 54415455 DB 'TATU', 'S'+80H, 0EEH
63B2 D3EE
63B4 52D1ED DB 'R', 'Q'+80H, 0EDH
63B7 00 DB 0
63B8 524FCEA2 TOPT: DB 'RO', 'N'+80H, 0A2H
63BC 524F46C6 DB 'ROF', 'F'+80H, 0A3H
63C0 A3
63C1 4142A8D9 DB 'AB', 'C'+80H, 0D9H
63C5 CFD7 DB 'D'+80H, 0D7H
63C7 48454E44 DB 'HEND', 'O'+80H, 0BEH
63CB CFBE
63CD 4845CED8 DB 'HE', 'N'+80H, 0D8H
63D1 41CE0D DB 'A', 'N'+80H, 0DH
63D4 494D45A4 DB 'IME', 'S'+80H, 0EAH
63D8 EA
63D9 00 DB 0
63DA 53494EC7 TOPU: DB 'SIN', 'G'+80H, 0E2H

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 8

```

63DE E2
63DF 53D2DB      DB 'S','R'+80H,0DBH
63E2 00          DB 0

63E3 41CC14      TOPV: DB 'A','L'+80H,14H
63E6 41525054    DB 'ARPT','R'+80H,0E5H
63EA D2E5
63EC 00          DB 0

63ED 494454CB    TOPW: DB 'IDT','H'+80H,0A0H
63F1 A0
63F2 4149D496    DB 'A1','T'+80H,96H
63F6 00          DB 0

63F7 4FD2FA      TOPX: DB 'O','R'+80H,0FAH
63FA 00          DB 0

63FB 00          TOPY: DB 0

63FC 00          TOPZ: DB 0

63FD ABF3        SPLTBL:DB '+'80H,0F3H
63FF ADF4        DB '-'80H,0F4H
6401 AAF5        DB 'X'+80H,0F5H
6403 AFF6        DB '/'80H,0F6H
6405 DEF7        DB '^'+80H,0F7H
6407 DCFE        DB 0DCH,0FEH
6409 A2F4        DB "'"+80H,0F4H
640B BEF0        DB '>'+80H,0F8H
640D BDF1        DB '='80H,0F1H
640F BCF2        DB '<'+80H,0F2H

6411 0079797C    PRIOTY:DB 0,79H,79H,7CH,7CH,7FH,50H,46H,3CH,32H,28H,7AH,7BH
6415 7C7F5046
6419 3C32287A
641D 7B

641E DF27        DATONV:DW 27DFH          ;CDBL
6420 0000        DW 0
6422 7F27        DW 277FH          ;CINT
6424 FA27        DW 27FAH          ;CHKSTR
6426 B327        DW 27B3H          ;CSNG

6428 CA29        DBLTBL:DW 29CAH          ;DBLADD
642A C329        DW 29C3H          ;DBLSUB
642C F42A        DW 2AF4H          ;DBLMLT
642E 372B        DW 2B37H          ;DBLDIV
6430 7827        DW 2778H          ;DCP

6432 1224        SNGTBL:DW 2412H          ;FADD
6434 0F24        DW 240FH          ;FSUB
6436 4125        DW 2541H          ;FLMULT
6438 9C25        DW 259CH          ;FLWDIV
643A 0C27        DW 270CH          ;FLOWCP

643C DD28        INTTBL:DW 28DDH          ;INTADD
643E D228        DW 28D2H          ;INTSUB
6440 FD28        DW 28FDH          ;INTMLT
6442 EA4B        DW 4BEAH          ;IDIVID
6444 3927        DW 2739H          ;ICP
***** END TABLE OF COMMAND *****

```



```

-----
BASIC ERROR MESSAGES TABLE
-----

```

```

6446 00      ERRMSG:DB 0
6447 4E455854 DB 'NEXT without FOR',0
6448 20776974
644F 686F7574
6453 20464F52
6457 00
6458 73796E74 DB 'syntax error',0
645C 617B2065
6460 72726F72
6464 00
6465 52455455 DB 'RETURN without GOSUB',0
6469 524E2077
646D 6974686F
6471 75742047
6475 4F535542
6479 00
647A 4F757420 DB 'Out of DATA',0
647E 6F662044
6482 41544100
6486 496C6C65 DB 'Illegal function call',0
648A 67616C20
648E 66756E63
6492 74696F6E
6496 2063616C
649A 6C00
649C 4F766572 DB 'Overflow',0
64A0 666C6F77
64A4 00
64A5 4F757420 DB 'Out of memory',0
64A9 6F66206D
64AD 656D6F72
64B1 7900
64B3 556E6465 DB 'Undefined line number',0
64B7 66696E65
64BB 64206C69
64BF 6E65206E
64C3 756D6265
64C7 7200
64C9 53756273 DB 'Subscript out of range',0
64CD 63726978
64D1 74206F75
64D5 74206F66
64D9 2072616E
64DD 676500
64E0 52656469 DB 'Redimensioned array',0
64E4 6D656E73
64E8 696F6E65
64EC 64206172
64F0 72617900
64F4 44697669 DB 'Division by zero',0
64F8 73696F6E
64FC 20627920
6500 7A65726F
6504 00
6505 496C6C65 DB 'Illegal direct',0
6509 67616C20

```


XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 10

650D 64697265	
6511 637400	
6514 54797065	DB 'Type mismatch',0
6518 206D6973	
651C 6D617463	
6520 6800	
6522 4F757420	DB 'Out of string space',0
6526 6F662073	
652A 7472696E	
652E 67207370	
6532 61636500	
6536 53747269	DB 'String too long',0
653A 6E672074	
653E 6F6F206C	
6542 6F6E6700	
6546 53747269	DB 'String formular too complex',0
654A 6E672066	
654E 6F726D75	
6552 6C617220	
6556 746F6F20	
655A 636F6D70	
655E 6C657000	
6562 43616E22	DB 'Can't continue',0
6566 7420636F	
656A 6E74696E	
656E 756500	
6571 556E6465	DB 'Undefined user function',0
6575 66696E65	
6579 64207573	
657D 65722066	
6581 756E6374	
6585 696F6E00	
6589 4E6F2052	DB 'No RESUME',0
658D 4553554D	
6591 4500	
6593 52455355	DB 'RESUME without error',0
6597 4D452077	
659B 6974686F	
659F 75742065	
65A3 72726F72	
65A7 00	
65A8 556E7072	DB 'Unprintable error',0
65AC 696E7461	
65B0 626C6520	
65B4 6572726F	
65B8 7200	
65BA 4D697373	DB 'Missing operand',0
65BE 696E6720	
65C2 6F706572	
65C6 616E6400	
65CA 4C696E65	DB 'Line buffer overflow',0
65CE 20627566	
65D2 66657220	
65D6 6F766572	
65DA 666C6F77	
65DE 00	
65DF 506F7369	DB 'Position not on screen',0
65E3 74696F6E	
65E7 206E6F74	
65EB 206F6E20	
65EF 73637265	

** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 11

```

65F3 656E00
65F6 42616420      DB  'Bad File Data',0
65FA 46696C65
65FE 20446174
6602 6100
6604 4469736B      DB  'Disk Basic Feature',0
6608 20426173
660C 69632046
6610 65617475
6614 726500
6617 436F6D6D      DB  'Communication Buffer Overflow',0
661B 756E6963
661F 6174696F
6623 6E204275
6627 66666572
662B 204F7665
662F 72666C6F
6633 7700
6635 506F7274      DB  'Port not initialized',0
6639 206E6F74
663D 20696E69
6641 7469616C
6645 697A6564
6649 00
664A 54617065      DB  'Tape read ERROR',0
664E 20726561
6652 64204552
6656 524F5200
665A 49465320      DB  'IFS without THENDO',0
665E 77697468
6662 6F757420
6666 5448454E
666A 444F00
666D 49465320      DB  'IFS without ELSEDO',0
6671 77697468
6675 6F757420
6679 454C5345
667D 444F00
6680 49465320      DB  'IFS without ENDIF',0
6684 77697468
6688 6F757420
668C 454E4449
6690 4600
6692 5448454E      DB  'THENDO without IFS',0
6696 444F2077
669A 6974686F
669E 75742049
66A2 465300
66A5 454C5345      DB  'ELSEDO without IFS',0
66A9 444F2077
66AD 6974686F
66B1 75742049
66B5 465300
66B8 454E4449      DB  'ENDIF without IFS',0
66BC 46207769
66C0 74686F75
66C4 74204946
66C8 5300
66CA 4D69732D      DB  'Mis-sequence',0
66CE 73657175
66D2 656E6365
66D6 206F6620
66DA 4946532D      DB  'IFS-THENDO-ELSEDO-ENDIF',0
66DE 5448454E
66E2 444F2D45
66E6 4C534544
66EA 4F2D454E
66EE 44494600

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 12

```

00
454E4444      DB  'ENDDO without THENDO or ELSEDO'
4F207769
74686F75
74205448
454E444F
206F7220
454C5345
444F
00
4C4F4F50      DB  0
20776974      DB  'LOOP without ENDLOOP',0
686F7574
20454E44
4C4F4F50
00
454E444C      DB  'ENDLOOP without LOOP',0
4F4F5020
77697468
6F757420
4C4F4F50
00
45584954      DB  'EXIT WHEN without LOOP',0
20574845
4E207769
74686F75
74204C4F
4F5000
4D756C74      DB  'Multi-defined label name',0
692D6465
66696E65
64206C61
62656C20
6E616D65
00
556E6465      DB  'Undefined label name',0
66696E65
64206C61
62656C20
6E616D65
00
52455455      DB  'RETURNL without GOSUBL',0
524E4C20
77697468
6F757420
474F5355
424C00
;
;-----
;
; LOCAL AND NEW COMMAND
;-----
;
0074  VARBGN:DW  07400H
0074  VAREND:DW 07400H
0074  ARYBGN:DW 07400H
0074  ARYEND:DW 07400H
;*****
;
;***** NEW STATEMENT FROM CALL HOOK 7 *****
E5    NEWLCL:PUSH HL
210074 LD  HL,07400H

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 13

```

227C67      LD  (VARGN),HL
227E67      LD  (VAREND),HL
228067      LD  (ARYBGN),HL
228267      LD  (ARYEND),HL
3E00        LD  A,0
325173      LD  (LCLON),A      ; SWITCH OFF LOCAL FLAG
325073      LD  (STKPTR),A
E1          POP HL
C9          RET
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;
2B          LOCAL0:DEC HL
D7          RST 10H
C8          RET Z
CD9B40      CALL 409BH
2C          DB  ',,'      ; 200 STATEMENT INC L XXXXXXXX
;
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;
F5          PUSH AF
3E01        LD  A,1
325173      LD  (LCLON),A      ; SWITCH ON LOCAL FLAG
F1          POP AF
019E67      LOCAL: LD  BC,LOCAL0
C5          PUSH BC
C3BF67      JP  GETVAR
;
;
2B          DIMLC0:DEC HL
D7          RST 10H
C8          RET Z
CD9B40      CALL 409BH
2C          DB  ',,'
;
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;
01B367      DIMLCL:LD  BC,DIMLC0
C5          PUSH BC
F6          DB  0F6H
;
;GET VARIABLE
;
AF          GETVAR1:XOR A
3244EF      LD  (0EF44H),A
46          LD  B,(HL)
CD9844      GTVAR1:CALL 449BH
DADF3B      JP  C,3BDFH
AF          XOR A
4F          LD  C,A
D7          RST 10H
3805        JR  C,GTVAR2
CD9944      CALL 4499H
3809        JR  C,GTVAR4
4F          GTVAR2:LD  C,A
D7          GTVAR3:RST 10H
38FD        JR  C,GTVAR3
CD9944      CALL 4499H
38F8        JR  NC,GTVAR3
110468      GTVAR4:LD  DE,GTVAR5
D5          PUSH DE
1602        LD  D,2

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 14

```

FE25      CP      '%'
C8        RET    Z
14        INC    D
FE24      CP      '0'
C8        RET    Z
14        INC    D
FE21      CP      '!'
C8        RET    Z
1608     LD      D,B
FE23      CP      '#'
C8        RET    Z
78        LD      A,B
D641     SUB     'A'
E67F     AND     7FH
5F        LD      E,A
1600     LD      D,B
E5        PUSH   HL
21A8EF   LD      HL,0EFABH
19        ADD    HL,DE
56        LD      D,(HL)
E1        POP    HL
2B        DEC    HL
C9        RET
7A        GTVAR5:LD  A,D
3245EF   LD      (0EF45H),A
D7        RST    10H
3A83EF   LD      A,(0EF83H)
3D        DEC    A
CA0469   JP      Z,GTARY4
F21E68   JP      P,GTVAR6
7E        LD      A,(HL)
D628     SUB     ''
CAD668   JP      Z,GETARY
D633     SUB     33H
CAD668   JP      Z,GETARY
AF        GTVAR6:XOR  A
3283EF   LD      (0EF83H),A
E5        PUSH   HL
D5        PUSH   DE
3A95F8   LD      A,(0F095H)
B7        OR     A
3292F8   LD      (0F092H),A
CA6768   JP      Z,GTVARA
2AC4EF   LD      HL,(0EFC4H)
11C6EF   LD      DE,0EFC6H.
19        ADD    HL,DE
227E67   LD      (VAREND),HL
EB        EX     DE,HL
EB        GTVAR7:EX  DE,HL
2A7E67   LD      HL,(VAREND)
CD9540   CALL   4095H
281A     JR      Z,GTVAR9
E1        POP    HL
1A        LD      A,(DE)
6F        LD      L,A
BC        CP     H
13        INC    DE
200B     JR      NZ,GTVARB
1A        LD      A,(DE)
B9        CP     C
2007     JR      NZ,GTVARB

```


XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 15

```

13          INC DE
1A          LD A,(DE)
B8          CP B
CAB768     JP Z,NEWVA2
3E          DB 3EH
13          GTVAR8:INC DE
13          INC DE
E5          PUSH HL
2600       LD H,0
19          ADD HL,DE
18DD       JR GTVAR7
3A92F0     GTVAR9:LD A,(0F092H)
B7          OR A
CA7348     JP Z,NEWVA2
AF          XOR A
3292F0     LD (0F092H),A
2A8067     GTVARA:LD HL,(ARYBGN)
227E67     LD (VAREND),HL
2A7C67     LD HL,(VARBGN)
C33968     JP GTVAR7

;
;CREATE NEW VARIABLE
;
F1          NEWVA1:POP AF
E1          POP HL
E3          EX (SP),HL
F5          PUSH AF
D5          PUSH DE
11594C     LD DE,4C59H
CD9540     CALL 4095H
283A       JR Z,NEWVA3
11BE4C     LD DE,4CBEH
CD9540     CALL 4095H
D1          POP DE
2837       JR Z,NEWVA4
F1          POP AF
E3          EX (SP),HL
E5          PUSH HL
C5          PUSH BC
4F          LD C,A
C5          PUSH BC
0600       LD B,0
03          INC BC
03          INC BC
03          INC BC
2A8267     LD HL,(ARYEND)
E5          PUSH HL
09          ADD HL,BC
C1          POP BC
E5          PUSH HL
CD8D38     CALL 388DH
E1          POP HL
228267     LD (ARYEND),HL
60          LD H,B
69          LD L,C
228067     LD (ARYBGN),HL
2B          NEWVA1:DEC HL
3600       LD (HL),0
CD9540     CALL 4095H
28F8       JR NZ,NEWVA1
D1          POP DE
;ARYBGN
;VAREND
;VARBGN
;ARYEND
;ARYBGN

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 16

```

73          LD  (HL),E
23          INC HL
D1          POP DE
73          LD  (HL),E
23          INC HL
72          LD  (HL),D
EB          EX  DE,HL
13          NEWA2:INC DE
E1          POP HL
C9          RET

;
;UNDEFINED VARIABLE FROM VAPTR
;
57          NEWA3:LD  D,A
5F          LD  E,A
F1          POP AF
F1          POP AF
E3          EX  (SP),HL
C9          RET

;
;UNDEFINED VARIABLE FROM FACTOR
;
32ABF8     NEWA4:LD  (8F0ABH),A
C1          POP BC
67          LD  H,A
6F          LD  L,A
22ABF8     LD  (8F0ABH),HL
CDD84D     CALL 4DD8H
2806       JR  NZ,NEWA5
215F3B     LD  HL,3B5FH
22ABF8     LD  (8F0ABH),HL
E1          NEWA5:POP HL
C9          RET

;
;GET ARRAY
;
E5          GETARY:PUSH HL
2A44EF     LD  HL,(0EF44H)
E3          EX  (SP),HL
57          LD  D,A
D5          GTARY1:PUSH DE
C5          PUSH BC
CDA844     CALL 44A8H
C1          POP BC
F1          POP AF
EB          EX  DE,HL
E3          EX  (SP),HL
E5          PUSH HL
EB          EX  DE,HL
3C          INC  A
57          LD  D,A
7E          LD  A,(HL)
FE2C       CP  ''
28EE       JR  Z,GTARY1
FE5D       CP  ''
2803       JR  NZ,GTARY2
D7         RST 10H
1804       JR  GTARY3

;
CD9B48     GTARY2:CALL 489BH
29         DB  ''

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 17

```

229AEF      |
E1          |   GTARY3:LD  (0EF9AH),HL
2244EF      |           POP  HL
1E00        |           LD  (0EF44H),HL
D5          |           LD  E,0
11          |           PUSH DE
E5          |           DB  11H
F5          |   GTARY4:PUSH HL
2A0067      |           PUSH AF
3E          |           LD  HL,(ARYB0N)
           |           DB  3EH

19          |   |
EB          |   GTARY5:ADD  HL,DE
2A0267      |           EX  DE,HL
EB          |           LD  HL,(ARYEND)
CD9540      |           EX  DE,HL
3A45EF      |           CALL 4095H
202A        |           LD  A,(0EF45H)
BE          |           JR  Z,MAKARY
23          |           CP  (HL)
2008        |           INC  HL
7E          |           JR  NZ,GTARY6
B9          |           LD  A,(HL)
23          |           CP  C
2004        |           INC  HL
7E          |           JR  NZ,GTARY7
B8          |           LD  A,(HL)
3E          |           CP  B
           |           DB  3EH

23          |   |
23          |   GTARY6:INC  HL
5E          |   GTARY7:INC  HL
23          |           LD  E,(HL)
56          |           INC  HL
23          |           LD  D,(HL)
20DE        |           INC  HL
3A44EF      |           JR  NZ,GTARY5
B7          |           LD  A,(0EF44H)
C2E03B      |           OR  A
F1          |           JP  NZ,3BE0H
44          |           POP  AF
4D          |           LD  B,H
C0A25       |           LD  C,L
96          |           JP  Z,250AH
CA9B69      |           SUB (HL)
1E09        |           JP  Z,GTARY0
C3F93B      |   BSERR:LD  E,9
           |           JP  03BF9H

77          |   |
23          |   MAKARY:LD  (HL),A
5F          |           INC  HL
1600        |           LD  E,A
F1          |           LD  D,0
CA0544      |           POP  AF
71          |           JP  Z,044A5H
23          |           LD  (HL),C
70          |           INC  HL
23          |           LD  (HL),B
4F          |           INC  HL
CD9D3B      |           LD  C,A
           |           CALL 03B9DH

```


XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 18

```

23          INC HL
23          INC HL
227BEF     LD (0EF7BH),HL
71         LD (HL),C
23          INC HL
3A44EF     LD A,(0EF44H)
17         RLA
79         LD A,C
010B00    MAKAR1:LD BC,0BH
3002       JR NC,MAKAR2
C1         POP BC
03         INC BC
71         MAKAR2:LD (HL),C
23         INC HL
70         LD (HL),B
23         INC HL
F5        PUSH AF
CDB528    CALL 20B5H
F1        POP AF
3D        DEC A
20ED      JR NZ,MAKAR1
F5        PUSH AF
42        LD B,D
4B        LD C,E
EB        EX DE,HL
19        ADD HL,DE
DAB43B    JP C,3BB4H
CDA43B    CALL 3BA6H
220267    LD (ARYEND),HL
2B        MAKAR3:DEC HL
3600      LD (HL),0
CD9540    CALL 4095H
20F8      JR NZ,MAKAR3
03        INC BC
57        LD D,A
2A7BEF    LD HL,(0EF7BH)
5E        LD E,(HL)
EB        EX DE,HL
29        ADD HL,HL
09        ADD HL,BC
EB        EX DE,HL
2B        DEC HL
2B        DEC HL
73        LD (HL),E
23        INC HL
72        LD (HL),D
23        INC HL
F1        POP AF
3032      JR C,GTARYC
47        GTARY8:LD B,A
4F        LD C,A
7E        LD A,(HL)
23        INC HL
14        DB 16H
E1        GTARY9:POP HL
5E        LD E,(HL)
23        INC HL
56        LD D,(HL)
23        INC HL
E3        EX (SP),HL
F5        PUSH AF

```


*** DUAD-PC ASSEMBLER (A.C.P.) ***

PAGE 19

```

CD9540      CALL 4095H
D23D69      JP  NC,BSERR
CDB528      CALL 2085H
19          ADD  HL,DE
F1          POP  AF
3D          DEC  A
44          LD   B,H
4D          LD   C,L
20E9        JR   NZ,GTARY9
3A45EF      LD   A,(0EF45H)
44          LD   B,H
4D          LD   C,L
29          ADD  HL,HL
D604        SUB  4H
3804        JR   C,GTARYA
29          ADD  HL,HL
2806        JR   Z,GTARYB
29          ADD  HL,HL
B7          GTARYA:OR  A
E2CA69      JP   PD,GTARYB
09          ADD  HL,BC

;
C1          GTARYB:POP BC
09          ADD  HL,BC
EB          EX   DE,HL
2A9AEF      GTARYC:LD  HL,(0EF9AH)
C9          RET
    
```

***** END LOCAL COMMAND *****

ERASE LOCAL VARIABLE ENTRY POINT

```

AF          ERASE1: XOR  A
3285EF      LD   (0EF85H),A      ;GR3
3E01        LD   A,1
3283EF      LD   (0EF83H),A      ;GTVPRM
CD3C4E      CALL 04E3CH        ;CALL GET VARIABLE
E5          PUSH HL
3283EF      LD   (0EF83H),A      ;GTVPRM
60          LD   H,B
69          LD   L,C
0B          DEC  BC
0B          DEC  BC
0B          DEC  BC
0B          DEC  BC
0B          DEC  BC
3A85EF      LD   A,(0EF85H)      ;GR3
B7          OR   A
201C        JR   NZ,10ARAY
19          ADD  HL,DE
EB          EX   DE,HL
2A8267      LD   HL,(ARYEND)    ;ARYEND *****
CD9540      ERASE1: CALL 4095H
1A          LD   A,(DE)
02          LD   (BC),A
13          INC  DE
03          INC  BC
    
```



```

20F7      JR    NZ,ERASE1
0B        DEC  BC
68        LD   H,B
69        LD   L,C
228267    LD   (ARYEND),HL      ;ARYEND *****
E1        POP  HL
7E        LD   A,(HL)
FE2C      CP   ,
C0        RET  NZ
D7        RST  10H
18C7      JR   ERASEL

```

```

F5        ; IOARY: PUSH AF
EB        EX   DE,HL
19        ADD  HL,DE
EB        EX   DE,HL
4E        LD   C,(HL)
0600      LD   B,0
09        ADD  HL,BC
09        ADD  HL,BC
23        INC  HL
06D2      LD   B,0D2H
FA226A    JP   M,IOARY1
78        LD   A,B
CDD70C    CALL 0CD7H
CDD70C    CALL 0CD7H
180B      JR   IOARY3

```

```

;
0E04      IOARY1:LD  C,4
CD880C    IOARY2:CALL 0C88H
B8        CP   B
20F8      JR   NZ,IOARY1
8D        DEC  C
20F7      JR   NZ,IOARY2
CDD84D    IOARY3:CALL 4DD8H
CAF73B    JP   Z,3BF7H
CD9540    IOARY4:CALL 4095H
CA476A    JP   Z,IOARY5
F1        POP  AF
F5        PUSH AF
7E        LD   A,(HL)
F4DA0C    CALL P,0CDAH
FC880C    CALL M,0C88H
77        LD   (HL),A
23        INC  HL
C3336A    JP   IOARY4
CD310C    IOARY5:CALL 0C31H
F1        POP  AF
E1        POP  HL
C9        RET

```

***** END ERASE COMMAND *****

----- GROUP OF IF AND LOOP COMMAND -----

```

;
F5        ; ***** IFS STATEMENT *****
C5        IF1: PUSH AF          ;SAVE CONDITION
D5        PUSH BC              ;X
E5        PUSH DE              ;X
CDA96A    PUSH HL              ;X
          CALL FNDADD          ;FIND ADDRESS OF THEN,ELSE,ENDIF

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 21

```

E1          POP HL          ;RETURN CONDITION
D1          POP DE          ;X
C1          POP BC          ;X
F1          POP AF          ;X
ED43676D   LD (GR1),BC     ;SAVE REGISTER BC
C1          POP BC          ;POP RETURN ADDRESS
ED43696D   LD (GR2),BC     ;SAVE RETURN ADDRESS IN GR2
ED4B736D   LD BC,(NIFADD)
C5          PUSH BC
;+++++++ PUSH LINE NUMBER OF ENDIF ++++++++
03          INC BC
03          INC BC
03          INC BC
22826D     LD (HLREG),HL
0A          LD A,(BC)
6F          LD L,A
03          INC BC
0A          LD A,(BC)
67          LD H,A
E5          PUSH HL          ;PUSH LINE NO. OF ENDIF
2A826D     LD HL,(HLREG)
;+++++++ END PUSH LINE NUMBER OF ENDIF ++++++
06BD      LD B,0BDH          ; LOAD TOKEN OF IFS
C5          PUSH BC          ; PUSH TOKEN OF IFS FOR CHECK
ED4B696D   LD BC,(GR2)
C5          PUSH BC          ;SET RETURN ADDRESS
ED4B676D   LD BC,(GR1)
C08F4A     CALL 4ABFH        ;EVALUATE EXPRESSION
E5          PUSH HL          ;X
CD9826     CALL 2690H        ;X FIND RESULT OF EVALUATE EXPRESSIO
E1          POP HL          ;X
CA936A     JP Z,IF3
2A6F6D     LD HL,(THNADD)   ;TRUE CONDITION
CD9A6A     CALL CHEXLN      ;CHANGE EXECLN TO LINE NO. OF THEN STAT
C9          RET
2A716D     IF3: LD HL,(ELSADD) ;FALSE CONDITION
CD9A6A     CALL CHEXLN      ;CHANGE EXECLN TO LINE NO. OF ELSE STAT
C9          RET
;          ***** END MAIN ROUTINE OF IF STATEMENT *****
;          ***** SUBROUTINE CHEXLN (CHANGE EXECLN ) *****
E5          CHEXLN: PUSH HL
23          INC HL
23          INC HL
23          INC HL
C5          PUSH BC
4E          LD C,(HL)
23          INC HL
46          LD B,(HL)
ED4352EB   LD (0EB52H),BC
C1          POP BC
E1          POP HL
C9          RET
;          ***** END SUBROUTINE CHEXLN *****
;          ***** SUBROUTINE OF FNDADD *****
;          FIND ADDRESS OF THEN,ELSE,ENDIF
;          ADDRESS OF THEN IS IN THNADD
;          ADDRESS OF ELSE IS IN ELSADD
;          ADDRESS OF ENDIF IS IN NIFADD
CDE06A     FNDADD: CALL SETFLG
CDBD6D     CALL NXTLIN      ;START NEXT LINE
;          X DO WHILE X AND Y AND Z = FALSE X

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 22

```

3A756D FND1: LD A,(X) ;
FE00 CP 0H
CAC76A JP Z,FND2
3A766D LD A,(Y)
FE00 CP 0H
CAC76A JP Z,FND2
3A776D LD A,(Z1)
FE00 CP 0H
C2DF6A JP NZ,ENDFND
CDB66D FND2: CALL ENDTXT ;END OF TEXT OR NOT
CDAD6D CALL GETTKN ;GET TOKEN
CD016B CALL TKNIF ;CHECK IF COMMAND
CD1F6B CALL TKNTHN ;CHECK THEN COMMAND
CD5D6B CALL TKNELS ;CHECK ELSE COMMAND
CD9B6B CALL TKNIF ;CHECK ENDIF COMMAND
CDDF6D CALL NDTLN1 ;NEXT LINE
C3AF6A JP FND1
ENDFND:RET
; ***** END SUBROUTINE OF FNDADD *****
; ***** SUBROUTINE OF SETFLG *****
F5 SETFLG:PUSH AF ;SAVE REG A FOR SET FLG
3E00 LD A,0H
32756D LD (X),A ;
32766D LD (Y),A ;X,Y,Z=FALSE
32776D LD (Z1),A ;
3E01 LD A,1H
32796D LD (FLGTHN),A ;
327A6D LD (FLGELS),A ;FLAG CHECKING FOR NESTING
327B6D LD (FLGNIF),A ;
32816D LD (FLGLP),A ;FLAG FOR LOOP COMMAND
3E03 LD A,3H
327C6D LD (FLGSYN),A
F1 POP AF
C9 RET
; ***** END SUBROUTINE OF SETFLG *****
; ***** SUBROUTINE FOR CHECKING IF COMMAND *****
3A786D TKNIF: LD A,(TOKEN)
FE0D CP 0BDH ;COMMAND WITH TOKEN OF IFS
C21E6B JP NZ,TKNIF2
3A796D TKNIF1:LD A,(FLGTHN) ;X
3C INC A ;X INC FLGTHN
32796D LD (FLGTHN),A ;X
3A7A6D LD A,(FLGELS) ;X
3C INC A ;X INC FLGELS
327A6D LD (FLGELS),A ;X
3A7B6D LD A,(FLGNIF) ;X
3C INC A ;X INC FLGNIF
327B6D LD (FLGNIF),A ;X
C9 TKNIF2:RET
; ***** END SUBROUTINE OF CHECKING IF COMMAND *****
; ***** SUBROUTINE OF CHECKING THEN COMMAND *****
3A786D TKNTHN:LD A,(TOKEN)
FE0E CP 0BEH ;COMPARE WITH TOKEN OF THEN
C2556B JP NZ,TKNTH2
3A796D LD A,(FLGTHN) ;X
3D DEC A ;X DEC FLGTHN
32796D LD (FLGTHN),A ;X
FE00 CP 0H ;COMPARE FLGTHN = 0 OR NOT
C2556B JP NZ,TKNTH2
CD566B CALL MTHADD ;MOVE THEN ADDRESS
3E01 LD A,1H ;X

```




** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 23

```

32796D      LD (FLGTH),A      ; SET FLGTH TO INITIAL CONDITION
32756D      LD (X),A         ;XSET X = TRUE
3A7C6D      LD A,(FLGSYN)    ;X
FE03        CP 3H           ;X CHECKING SEQUENCE
C2506B      JP NZ,TKNTH1    ;X
3A7C6D      LD A,(FLGSYN)    ;X
3D          DEC A           ;X DEC FLGSYN
327C6D      LD (FLGSYN),A   ;X
C3556B      JP TKNTH2
1E24        TKNTH1:LD E,24H  ;***** SEQUENCE ERROR *****
C3F93B      JP 3BF9H        ; JUMP TO ERROR ROUTINE AND RETURN B
C9          TKNTH2:RET
;          ***** END SUBROUTINE OF CHECKING THEN COMMAND **
;          ***** SUBROUTINE MTHADD *****
E5          MTHADD: PUSH HL
2B          DEC HL          ;SET POINTER TO POINT TO BYTE 00 BEFORE
226F6D      LD (THADD),HL
E1          POP HL
C9          RET
;          ***** END SUBROUTINE OF MTHADD *****
;          ***** SUBROUTINE OF CHECKING ELSE COMMAND *****
3A786D      TKNLS:LD A,(TOKEN)
FE0F        CP 0BFH        ;COMPARE WITH TOKEN OF ELSEDD
C2936B      JP NZ,TKNLS2
3A7A6D      LD A,(FLGELS)    ;X
3D          DEC A           ;X DEC FLGELS
327A6D      LD (FLGELS),A   ;X
FE00        CP 0H          ; COMPARE FLGELS = 0 OR NOT
C2936B      JP NZ,TKNLS2
CD946B      CALL MLSADD     ; MOVE ELSE ADDRESS
3E01        LD A,1H        ;XSET Y =TRUE =1
327A6D      LD (FLGELS),A   ;X
32766D      LD (Y),A       ;X
3A7C6D      LD A,(FLGSYN)    ;X
FE02        CP 2H          ;X CHECKING SEQUENCE
C28E6B      JP NZ,TKNLS1    ;X
3A7C6D      LD A,(FLGSYN)    ;X
3D          DEC A           ;X DEC FLGSYN
327C6D      LD (FLGSYN),A   ;X
C3936B      JP TKNLS2
1E24        TKNLS1:LD E,24H ;*****SEQUENCE ERROR *****
C3F93B      JP 3BF9H        ;JUMP TO ERROR ROUTINE AND RETURN TO
C9          TKNLS2:RET
;          ***** END SUBROUTINE OF CHECKING ELSEDD *****
;          ***** SUBROUTINE MLSADD *****
E5          MLSADD: PUSH HL
2B          DEC HL          ;SET POINTER TO POINT TO BYTE 00 BEFORE
22716D      LD (ELSDADD),HL ;
E1          POP HL
C9          RET
;          ***** END SUBROUTINE MLSADD *****
;          ***** SUBROUTINE OF CHECKING ENDIF COMMAND *****
3A786D      TKNNIF:LD A,(TOKEN)
FEC1        CP 0C1H        ;TOKEN OF ENDIF
C2D16B      JP NZ,TKNNI2
3A7B6D      LD A,(FLGNIF)    ;X
3D          DEC A           ;X DEC FLGNIF
327B6D      LD (FLGNIF),A   ;X
FE00        CP 0H          ;COMPARE FLGNIF = 0 OR NOT
C2D16B      JP NZ,TKNNI2
CDD26B      CALL MNFADD     ;MOVE ENDIF ADDRESS

```



```

3E01          LD  A,1H          ;X
327B6D        LD  (FLGNIF),A    ;SET FLGNIF TO INITIAL CONDITION
32776D        LD  (Z1),A       ;XSET.Y =TRUE=1
3A7C6D        LD  A,(FLGSYN)   ;X
FE01          CP   1H          ;X CHECKING SEQUENCE
C2CC6B        JP  NZ,TKNNI1    ;X
3A7C6D        LD  A,(FLGSYN)   ;X
3D            DEC  A           ;X DEC  FLGSYN
327C6D        LD  (FLGSYN),A   ;X
C3D16B        JP  TKNNI2
1E24          TKNNI1:LD  E,24H   ;XSEQUENCE ERROR XXXXXXXXXXXXXXXXXXXX
C3F93B        JP  3BF9H        ;JUMP TO ERROR ROUTINE AND RETURN TO
C9            TKNNI2:RET
;            ;XXXXXXXXXX END SUBROUTINE OF CHECKING ENDIF COMMAND XXXXXXXXX
;            ;XXXXXXXXXX SUBOUTINE MNFADD XXXXXXXXX
E5            MNFADD:PUSH HL
2B            DEC  HL          ;SET POINTER TO POINT TO BYTE 00 BEFORE
22736D        LD  (NIFADD),HL
E1            POP  HL
C9            RET
;            ;XXXXXXXXXX END MNFADD XXXXXXXXX
;            ;XXXXXXXXXX ENDDO STATEMENT XXXXXXXXX
ENDDO:
;----- CHECK SYNTAX ERROR -----
C0            RET  NZ
;            PUSH  HL
;            INC  HL
;            LD   A,(HL)
;            CP   00
;            LD   E,2
;            JP  NZ,3BF9H
;            POP  HL
;----- END CHECK SYNTAX ERROR -----
ED43676D      LD  (GR1),BC      ;SAVE BC REGISTER IN GR1
C1            POP  BC          ;SAVE RET ADD IN BC REGISTER
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
F1            POP  AF          ;POP TOKEN OF IFS
FE0D          CP   0BDH        ;COMPARE WITH TOKEN OF IFS
1E25          LD  E,25H        ;DISPLAY "ENDDO WITHOUT THENDO OR ELSE"
C2F93B        JP  NZ,3BF9H     ;JUMP TO DISPLAY ERROR
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
E1            POP  HL          ;X
2252EB        LD  (0EB52H),HL  ;XSET EXECLN = LINE NO. OF ENDIF
E1            POP  HL          ;SET HL = ENDIF ADDRESS
22696D        LD  (GR2),HL    ;CHECK ENDIF ADDRESS
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3EBD          LD  A,0BDH      ;PUSH TOKEN OF IFS FOR CHECK IN ENDI
F5            PUSH AF
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C5            PUSH BC        ;PUSH RET ADDRESS
ED4B676D      LD  BC,(GR1)    ;SET BC TO ORIGINAL VALUE
C9            RET
;            ;XXXXXXXXXX END ENDDO STATEMENT XXXXXXXXX
;            ;XXXXXXXXXX THENDO STATEMENT XXXXXXXXX
THENDO:
;-----CHECK SYNTAX ERROR -----
C0            RET  NZ
;            PUSH  HL
;            INC  HL
;            LD   A,(HL)
;            CP   00

```



```

; LD E,2
; JP NZ,3BF9H
; POP HL
;----- END CHECK SYNTAX ERROR -----
ED43676D LD (GR1),BC ;SAVE BC IN GR1
C1 POP BC ;POP RET ADD
F1 POP AF ;POP TOKEN OF IFS
FEBD CP 0BDH ;COMP WITH TOKEN OF IFS
1E21 LD E,21H ;DISPLAY "THENDO WITHOUT IFS"
C2F93B JP NZ,3BF9H ;JUMP DISPLAY ERROR AND RET TO BASIC
F5 PUSH AF ;PUSH TOKEN OF IFS
C5 PUSH BC ;PUSH RET ADD
ED4B676D LD BC,(GR1) ;RETURN VALUE OF BC
C9 RET
; ***** END THENDO STATEMENT *****
; ***** ELSEDO STATEMENT *****
ELSEDO:
;----- CHECK SYNTAX ERROR -----
C0 RET NZ
; PUSH HL
; INC HL
; LD A,(HL)
; CP 00
; LD E,2
; JP NZ,3BF9H
; POP HL
;----- END CHECK SYNTAX ERROR -----
ED43676D LD (GR1),BC ;SAVE BC IN GR1
C1 POP BC ;POP RET ADD
F1 POP AF ;POP TOKEN OF IFS
FEBD CP 0BDH ;COMP WITH TOKEN OF IFS
1E22 LD E,22H ;DISPLAY "ELSEDO WITHOUT IFS"
C2F93B JP NZ,3BF9H ;JUMP TO DISP ERROR AND RETURN TO BA
F5 PUSH AF ;PUSH TOKEN OF IFS
C5 PUSH BC ;PUSH RET ADD
ED4B676D LD BC,(GR1) ;RETURN VALUE OF BC
C9 RET
; ***** END ELSEDO STATEMENT *****
; ***** ENDIF STATEMENT *****
ENDIF:
;----- CHECK SYNTAX ERROR -----
C0 RET NZ
; PUSH HL
; INC HL
; LD A,(HL)
; CP 00
; LD E,2
; JP NZ,3BF9H
; POP HL
;----- END CHECK SYNTAX ERROR -----
ED43676D LD (GR1),BC ;SAVE BC IN GR1
C1 POP BC ;POP RET ADD
F1 POP AF ;POP TOKEN OF IFS
FEBD CP 0BDH ;COMP WITH TOKEN OF IFS
1E23 LD E,23H ;DISPLAY "ENDIF WITHOUT IFS"
C2F93B JP NZ,3BF9H ;JUMP TO DISPLAY ERROR AND RETURN TO
C5 PUSH BC ;PUSH RETURN ADD
ED4B676D LD BC,(GR1) ;RETURN VALUE OF BC
C9 RET
; ***** END ENDIF STATEMENT *****

```



```

XXXXXXXXXXXXXXXXX END GROUP OF IF STATEMENT XXXXXXXXXXXXXXX
|-----
|                               GROUP OF IF AND LOOP COMMAND
|-----
|                               XXXXXXXXXXXX LOOP STATEMENT XXXXXXXXXXXX
LOOP:
|----- CHECK SYNTAX ERROR -----
C0      RET    NZ
|      PUSH   HL
|      INC    HL
|      LD     A,(HL)
|      CP     00
|      LD     E,2
|      JP     NZ,3BF9H
|      POP    HL
|----- END CHECK SYNTAX ERROR -----
22826D  LD     (HLREG),HL      ;SEE INPUT POINTER
F5      LOOP1: PUSH  AF
C5      PUSH  BC
D5      PUSH  DE
E5      PUSH  HL
CDBA6C  CALL  MLPADD          ;MOVE LOOP ADDRESS
E1      POP   HL
D1      POP   DE
C1      POP   BC
F1      POP   AF
F5      PUSH  AF
C5      PUSH  BC
D5      PUSH  DE
E5      PUSH  HL
CDB66C  CALL  FNLADD          ;FIND END LOOP ADDRESS
E1      POP   HL
D1      POP   DE
C1      POP   BC
F1      POP   AF
ED4367D LD     (GR1),BC          ;SAVE BC REGISTER IN GR1
C1      POP   BC              ;SAVE RET ADDRESS IN BC REGISTER
22696D  LD     (GR2),HL        ;SAVE HL REGISTER IN GR2
2A7D6D  LD     HL,(NLPADD)     ;*
E5      PUSH  HL              ;* PUSH END LOOP ADDRESS
;+++++ PUSH LINE NUMBER OF ENDL0OP STATEMENT +++++
23      INC   HL
23      INC   HL
23      INC   HL
ED536C78 LD   (DE1),DE *
5E      LD   E,(HL)
23      INC   HL
56      LD   D,(HL)
D5      PUSH  DE              ;PUSH LINE NO. OF ENDL0OP STATEMENT
ED5B6C78 LD   DE,(DE1)
;+++++ END PUSH LINE NUMBER OF ENDL0OP STATEMENT +++++
2A7F6D  LD   HL,(LOPADD)      ;*
E5      PUSH  HL              ;* PUSH LOOP ADDRESS
;+++++ PUSH LINE NUMBER OF LOOP STATEMENT +++++
23      INC   HL
23      INC   HL
23      INC   HL
ED536C78 LD   (DE1),DE
5E      LD   E,(HL)
23      INC   HL
56      LD   D,(HL)

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 27

```

D5          PUSH DE
;+++++++ END PUSH LINE NUMBER OF LOOP STATEMENT ++++++++
168B        LD D,0BBH          ;LOAD TOKEN OF LOOP STATEMENT
D5          PUSH DE
ED5B6C70   LD DE,(DE1)
C5          PUSH BC          ;PUSH RET ADDRESS
ED4B676D   LD BC,(GR1)      ;X
2A696D     LD HL,(GR2)      ;X SET BC,HL
C9          RET
;          ***** END LOOP STATEMENT *****
;          ***** SUBROUTINE MLPADD *****
E5          MLPADD:PUSH HL
2B          MLPAD1:DEC HL
7E          LD A,(HL)        ;X
FE20        CP 20H          ;X SKIP SPACE
CA8B6C     JP Z,MLPAD1      ;X
C5          PUSH BC
ED4B53EB   LD BC,(0EB53H)
B9          CP C
C1          POP BC
C28B6C     JP NZ,MLPAD1
2B          DEC HL
7E          LD A,(HL)
C5          PUSH BC
ED4B52EB   LD BC,(0EB52H)
B9          CP C
C1          POP BC
C28B6C     JP NZ,MLPAD1
C5          PUSH BC
2B          DEC HL
46          LD B,(HL)
2B          DEC HL
4E          LD C,(HL)
68          LD H,B
69          LD L,C
C1          POP BC
2B          DEC HL          ;SET POINTER TO POINT TO BYTE 00 BEFORE
227F6D     LD (LOPADD),HL
E1          POP HL
C9          RET
;          ***** END SUBROUTINE MLPADD *****
;          ***** SUBROUTINE FNLADD *****
CDE06A     FNLADD:CALL SETFLG
CDBD6D     CALL NXTLIN
;          ***** DO WHILE *****
3A756D     FNLAD1:LD A,(X)
FE00        CP 0H          ; 0H = FALSE
C2D66C     JP NZ,FNLAD2
CDF46D     CALL NTXTLOP
CDAD6D     CALL GETTKN
CDD76C     CALL TKNLNP
CDE76C     CALL TKNLNP     ; FIND ENDLOOP COMMAND
CDDF6D     CALL NXTLN1    ; NEXT LINE
C3BC6C     JP FNLAD1
C9          FNLAD2:RET
;          ***** END SUBROUTINE FNLADD *****
;          ***** SUBROUTINE TKNLNP *****
3A7B6D     TKNLNP:LD A,(TOKEN)
FE00        CP 0BBH        ; TOKEN OF LOOP
C2E66C     JP NZ,TKNLP1
3A816D     LD A,(FLGLP)    ;X

```



```

3C          INC A          ;X INC FLGLP
32816D     LD (FLGLP),A   ;X
C9          TKNLP:RET
;          ;XXXXXXXXX END SUBROUTINE TKNLOP XXXXXXXX
;          ;XXXXXXXXX SUBROUTINE TKNLNP XXXXXXXX
3A786D     TKNLNP:LD A,(TOKEN)
FEBC      CP 0BCH          ; TOKEN OF ENDOOP
C2866D     JP NZ,TKNLI
3A816D     LD A,(FLGLP)   ;X
3D         DEC A          ;X DEC FLGLP
32816D     LD (FLGLP),A   ;X
FE80      CP 0H
C2866D     JP NZ,TKNLI
CD876D     CALL MNLADD
3E81      LD A,1H
32816D     LD (FLGLP),A   ;X SET FLGLP TO INITIAL CONDITION
32756D     LD (X),A
C9          TKNLI:RET
;          ;XXXXXXXXX END SUBROUTINE TKNLNP XXXXXXXX
;          ;XXXXXXXXX SUBROUTINE MNLADD XXXXXXXX
;          ;THIS ROUTINE MOVE ADDRESS OF STATEMENT AFTER
;          ;END LOOP STATEMENT TO NLPADD
;          ;(NLPADD POINT TO BYTE 00 BEFORE HEADING OF LINE
;          ;STATEMENT)
E5          MNLADD:PUSH HL
C5         PUSH BC
4E         LD C,(HL)
23         INC HL
46         LD B,(HL)
68         LD H,B
69         LD L,C
2B         DEC HL          ;SET POINTER TO POINT TO BYTE 00 BEFORE
227D6D     LD (NLPADD),HL
C1         POP BC
E1         POP HL
C9         RET
;          ;XXXXXXXXX END SUBROUTINE MNLADD XXXXXXXX
-----
;          ;WHEN STATEMENT
-----
ED43676D   WHEN: LD (GR1),BC ;SAVE BC IN GR1
C1         POP BC          ;POP RET ADD
F1         POP AF          ;POP TOKEN OF LOOP
FE8B      CP 0BBH         ;COMP WITH TOKEN OF LOOP
1E28      LD E,28H        ;DISPLAY "EXIT WHEN WITHOUT LOOP"
C2F93B     JP NZ,3BF9H    ;JUMP TO ERROR ROUTINE AND RET TO BA
F5        PUSH AF         ;PUSH TOKEN OF LOOP
C5        PUSH BC         ;PUSH RET ADD
ED4B676D   LD BC,(GR1)    ;RET VALUE OF BC
-----
C8BF4A    CALL 4ABFH      ;EVALUATE EXPRESSION
E5        PUSH HL        ;X
CD9826    CALL 2698H      ;X FIND RESULT OF EVALUATE EXPRESSION
E1        POP HL         ;X
CA456D    JP Z,WHEN1
ED43676D   LD (GR1),BC
C1        POP BC
F1        POP AF
E1        POP HL
;          ;SAVE RET ADDRESS IN BC
;          ;POP TOKEN OF LOOP
;          ;POP LINE NO. OF LOOP STATEMENT

```



```

E1          POP HL          ; POP LOOP ADDRESS
E1          POP HL          ; POP LINE NO. OF ENDLOOP STATEMENT
2252EB     LD (0EB52H),HL   ; CHANGE EXECLN TO LINE NO. OF ENDLOO
E1          POP HL          ; SET HL= ADD NEXT STATEMENT AFTER ENDLO
C5          PUSH BC         ; SET ORIGINAL RET ADDRESS
ED4B676D   LD BC,(GR1)
C9          WHEN1: RET
            ; ***** END WHEN STATEMENT *****
            ;-----
            ; ENDLOOP STATEMENT
            ;-----
            ; CHECK SYNTAX ERROR
C8          RET NZ
            ;
            ; PUSH HL
            ; INC HL
            ; LD A,(HL)
            ; CP 00
            ; LD E,2
            ; JP NZ,3BF9H
            ; POP HL
            ;-----
            ; END CHECK SYNTAX ERROR
ED43676D   ENLDP: LD (GR1),BC
ED536670   LD (GR6),DE
C1          POP BC          ; SAVE RET ADD IN GR1
F1          POP AF          ; POP TOKEN OF LOOP
FE8B      CP 0BBH          ; COMP WITH TOKEN OF LOOP
1E27      LD E,27H         ; DISP "ENDLOOP WITHOUT LOOP"
C2F93B     JP NZ,3BF9H     ; JUMP TO DISP ERROR AND RET TO BASIC
D1          POP DE          ; POP LINE NO. OF LOOP STATEMENT
ED5352EB   LD (0EB52H),DE  ; CHANGE EXECLN TO LINE NO. OF LOOP 6
E1          POP HL          ; SET HL = LOOP ADDRESS
E3          PUSH HL        ; PUSH LOOP ADD FOR NEXT TIME
D5          PUSH DE        ; PUSH LINE NO. OF LOOP STATEMENT FOR NE
F5          PUSH AF        ; PUSH TOKEN OF LOOP
C5          PUSH BC        ; PUSH RET ADD
ED4B676D   LD BC,(GR1)
C9          RET
            ; ***** END ENDLOOP STATEMENT *****
            ; ***** DEFINE STORAGE *****
0000      GR1: DW 0H
0000      GR2: DW 0H
0000      GR3: DW 0H
0000      GR4: DW 0H
0000      THNADD:DW 0H
0000      ELSADD:DW 0H
0000      NIFADD:DW 0H
00        X: DB 0H
00        Y: DB 0H
00        Z1: DB 0H
00        TOKEN: DB 0H
01        FLGTHN:DB 1H
01        FLGELS:DB 1H
01        FLGNIF:DB 1H
03        FLGSYN:DB 3H
0000      NLPADD:DW 0H
0000      LOPADD:DW 0H
01        FLGLP: DB 1H
0000      HLREG: DW 0H
00A7      COUNT: DW 0A700H

```



```

; ***** END DEFINE STORAGE *****
; ***** SUBROUTINE FINE END OF TEXT OR NOT *****
E5 ENDTXT: PUSH HL
7E LD A,(HL)
23 INC HL
B6 OR (HL)
C2AB6D JP NZ,ENDTX1
;*****
3A756D LD A,(X)
FE00 CP 0
1E1E LD E,1EH ;DISPLAY "IFS WITHOUT THENDO"
CAF93B JP Z,3BF9H ;JUMP TO DISP ERROR AND RET TO BASIC
3A766D LD A,(Y)
FE00 CP 0
1E1F LD E,1FH ;DISPLAY "IFS WITHOUT ELSEDO"
CAF93B JP Z,3BF9H ;JUMP TO DISP ERROR AND RET TO BASIC
3A776D LD A,(Z1)
FE00 CP 0
1E20 LD E,20H ;DISPLAY "IFS WITHOUT ENDF"
CAF93B JP Z,3BF9H ;JUMP TO DISP ERROR AND RET TO BASIC
E1 ENDTX1: POP HL
C9 RET
; ***** END SUBROUTINE FOR FINE END OF TEXT *****
; ***** SUBROUTINE GET TOKEN *****
E5 GETTKN: PUSH HL
23 INC HL
23 INC HL
23 INC HL
23 GETTKN1: INC HL
7E LD A,(HL)
FE20 CP 20H ;X SKIP SPACE
CAB16D JP Z,GETTKN1 ;X
32786D LD (TOKEN),A
E1 POP HL
C9 RET
; ***** END SUBROUTINE GET TOKEN *****
; ***** SUBROUTINE NEXT LINE *****
2B NXTLIN: DEC HL ;X
7E LD A,(HL) ;X
FE20 CP 20H ;X SKIP SPACE
CABD6D JP Z,NXTLIN ;X
C5 PUSH BC
ED4B53EB LD BC,(0EB53H)
B9 CP C ;X
C1 POP BC
C2BD6D JP NZ,NXTLIN ;XX FIND LINE NUMBER WITH EXECUTING
2B DEC HL ;X
7E LD A,(HL) ;X
C5 PUSH BC
ED4B52EB LD BC,(0EB52H)
B9 CP C ;X
C1 POP BC
C2BD6D JP NZ,NXTLIN ;X
2B DEC HL ;X
2B DEC HL ;X MOVE TO HEADER OF LINE
C3E06D JP NXTLN2 ;X
00 NXTLN1: NOP
4E NXTLN2: LD C,(HL) ;X
23 INC HL ;X
46 LD B,(HL) ;XX HL = HEADER OF NEXT LINE
60 LD H,B ;X

```



```

69          LD L,C          ;X
ED4B846D   LD BC,(COUNT) ;X
7D          LD A,L         ;X
82          LD (BC),A      ;X
83          INC BC         ;X TRACE HEADER NEXT LINE
7C          LD A,H         ;X
82          LD (BC),A      ;X
83          INC BC         ;X
ED43846D   LD (COUNT),BC ;X
C9          RET

;XXXXXXXX END SUBROUTINE NEXT LINE NUMBER XXXXXXXX
E5          NTXTLOPPUSH HL
23          INC HL
B6          OR (HL)
C2FF6D     JP NZ,NTXT1
1E26       LD E,26H        ;DISPLAY "LOOP WITHOUT ENLOOP"
C3F93B     JP 03BF9H
E1          NTXT1: POP HL
C9          RET

;XXXXXXXXXX END GROUP OF LOOP STATEMENT XXXXXXXXXXXXXXXXXXXX
;-----
;
; GOSUB LABEL COMMAND
;-----
CD776E     CALL INIFLG
CD886E     CALL SRET        ;PUSH ADD OF NEXT STATEMENT FOR RETURNL
7E          GOSUBL:LD A,(HL) ;X
FE28       CP 20H         ;X HL REG SKIP BLANK BETWEEN GOSUBL AND
C2116E     JP NZ,GOS1     ;X LABEL
23          INC HL
C3876E     JP GOSUBL
228A70     GOS1: LD (STALB),HL ;SAVE START ADD OF LABEL IN STALB
ED538678   LD (GR6),DE    ;SAVE DE REG IF GR6 #ONLY ONE LEVEL
ED5854EB   LD DE,(0EB54H) ;SET DE TO POINT TO START OF TEXT
CD886F     CALL SAVENT    ;CALL SAVE ADD OF NEXT LLINE TO BE C
13         INC DE        ;X
13         INC DE        ;X
13         INC DE        ;X SKIP HEAD OF THAT LINE
13         INC DE        ;X
CD8C6F     CALL DESKBI    ;DE REG DKIP BLANK BYY INCREMENT
1A          NOTEOF:LD A,(DE) ;LOAD TOKEN TO BE COMPARE
FEC2       CP 0C2H       ;COMPARE WITH TOKEN OF REM OR
CA3A6E     JP Z,COMP
CDA26F     CALL NEXTLG    ;MOVE DE TO NEXT LINE SKIP HEAD AND B
3A866E     LD A,(EOFTXT) ;X
FE00       CP 0          ;XCHECK IF END OF TEXT JUMP TO TRAN
CA4F6E     JP Z,TRAN     ;X
C3266E     JP NOTEOF
13         COMP: INC DE
CD8C6F     CALL DESKBI
CD486F     CALL COMSTR   ;X
CDA26F     CALL NEXTLG
3A866E     LD A,(EOFTXT) ;X
FE00       CP 0          ;XCHECK IF END OF TEXT JUMP TO TRAN
CA4F6E     JP Z,TRAN     ;X
C3266E     JP NOTEOF
3A876E     TRAN: LD A,(FLGLB) ;FLGLB SHOW STRING IS FOUND OR NOT
FE00       CP 0          ;FLGLB = 0 NOT FOUND ; FLGLB (>) 0 FOUND
CA656E     JP Z,ERROR1  ;IF STRING NOT FOUND JUMP TO ERROR
3A856E     TRAN1: LD A,(CHK1LB) ;X

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 32

```

3D          DEC A          ;CHECK LABEL IS MULTIPLE LABEL OR N
FE00       CP 0           ;X
CA6A6E     JP 2,LBOK      ;X
;+++++ CALL ERROR ++++++
1E2F       LD E,29H       ;DISP "MULTI-DEFINED LABEL NAME"
C3F93B     JP 3BF9H       ;JUMP TO DISP ERROR AND RET TO BASIC
1E2A       ERROR1:LD E,2AH ;DISP "UNDEFINED LABEL NAME"
C3F93B     JP 3BF9H       ;JUMP TO DISP ERROR AND RET TO BASIC
;+++++ END CALL ERROR ++++++
2A0C70     LBOK: LD HL,(CURLN) ;X
3E00       LD A,0         ;X
32856E     LD (CHKMLB),A   ; SET CHKMLB TO INITIAL CONDITION
ED5B0670   LD DE,(GR6)
C9         RET
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3E00       INIFLG:LD A,0
32856E     LD (CHKMLB),A
32876E     LD (FLGLB),A
3E01       LD A,1
32866E     LD (EOFTXT),A
C9         RET
00         CHKMLB:DB 0     ; FLAG USE TO CHECK MULTIPLE LABEL OR N
01         EOFTXT:DB 1    ; FLAG USE TO CHECK END OF TEXT OR NOT
00         FLGLB:DB 0     ; FLAG USE TO CHECK STRING IS FOUND OR
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
22826D     SRET: LD (HLREG),HL ;SAVE HL REG IN HLREG(HL POINT TO T
2B         SRET1: DEC HL    ;DEC FROM POINT TO TOKEN
7E         LD A,(HL)       ;X
FE20       CP 20H         ;X SKIP BLANK BY DECREMENT HL
CA8B6E     JP Z,SRET1      ;X
C5         PUSH BC        ;X
ED4B53EB   LD BC,(0EB53H) ;X
B9         CP C           ;X
C1         POP BC         ;X
C28B6E     JP NZ,SRET1    ;X
2B         DEC HL         ;X MOVE HL TO HEASER OF LINE
7E         LD A,(HL)      ;X HL = POINT HEADER OF LINE
C5         PUSH BC        ;X (HL)= POINT HEADER OF NEXT LINE
ED4B52EB   LD BC,(0EB52H) ;X
B9         CP C           ;X
C1         POP BC         ;X
C28B6E     JP NZ,SRET1    ;X
2B         DEC HL         ;X
2B         DEC HL         ;X
4E         SRET2: LD C,(HL) ;X
23         INC HL         ;X
46         LD B,(HL)      ;X
0B         DEC BC         ;DEC BC BECAUSE POINTER MUST POINT TO
;HEADER OF THAT STATEMENT WHEN RET TO THAT STATEMEN
E1         POP HL         ;X POP ADD OF RETURN TO MAIN PROGRAM
F1         POP AF         ;X POP ADD OF RETURN TO SYSTEM
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ED436A70   LD (HL1),BC    ;SAVE BC IN HL1 (BC POINT TO NEXT S
03         INC BC
03         INC BC
03         INC BC
E5         PUSH HL
60         LD H,B
69         LD L,C

```


DUAD-PC ASSEMBLER (A.C.P.)

PAGE 33

```

4E          LD  C,(HL)
23          INC HL
46          LD  B,(HL)
ED436870   LD  (EXECLN),BC          ;SAVE LINE NO. AFTER GOSUBL IN EXECL
;----- PUSH ADDRESS OF VARIABLE TABLE INTO TEMPORY STACK -----
C5          PUSH BC
DDE5       PUSH IX
F5          PUSH AF
DD2A4E73   LD  IX,(BASESK)
3A5073     LD  A,(STKPNT)
4F         LD  C,A
0600       LD  B,0
DD09       ADD  IX,BC
ED4B7C67   LD  BC,(VARBGN)
DD7100     LD  (IX),C
DD23       INC  IX
DD7000     LD  (IX),B
DD23       INC  IX
ED4B7E67   LD  BC,(VAREND)
DD7100     LD  (IX),C
DD23       INC  IX
DD7000     LD  (IX),B
DD23       INC  IX
ED4B8067   LD  BC,(ARYBGN)
DD7100     LD  (IX),C
DD23       INC  IX
DD7000     LD  (IX),B
DD23       INC  IX
ED4B8267   LD  BC,(ARYEND)
DD7100     LD  (IX),C
DD23       INC  IX
DD7000     LD  (IX),B
C600       ADD  A,B
325073     LD  (STKPNT),A
F1         POP  AF
DDE1       POP  IX
C1         POP  BC
;-----END PUSH ADDRESS OF VARIABLE TABLE INTO TEMPORY STACK ---
E1         POP  HL
ED4B7C67   LD  BC,(VARBGN)          ;+
C5         PUSH BC                  ;+ PUSH ADDRESS OF VARBGN IN STACK
ED4B7E67   LD  BC,(VAREND)         ;+
C5         PUSH BC                  ;+ PUSH ADDRESS OF VAREND IN STACK
ED4B8067   LD  BC,(ARYBGN)         ;+
C5         PUSH BC                  ;+ PUSH ADDRESS OF ARYBGN IN STACK
ED4B8267   LD  BC,(ARYEND)         ;+
C5         PUSH BC                  ;+ PUSH ADDRESS OF ARYEND IN STACK
ED4B6A70   LD  BC,(HL1)            ;X
C5         PUSH BC                  ;X BC POINT TO NEXT STATEMENT AFTER
ED4B6870   LD  BC,(EXECLN)         ;BC POINT LINE NO. OF NEXT STATEMENT
C5         PUSH BC
0600       LD  B,060H
C5         PUSH BC                  ;PUSH TOKEN OF GOSUBL
F5         PUSH AF                  ;XPUSH RETURN TO SYSTEM
E5         PUSH HL                  ;X PUSH RETURN TO MAIN PROGRAM
ED4B8267   LD  BC,(ARYEND)         ;X
ED437C67   LD  (VARBGN),BC         ;X
ED437E67   LD  (VAREND),BC         ;XSET AREA FOR NEW LOCAL VARIABLE
ED438067   LD  (ARYBGN),BC         ;X
2A826D     LD  HL,(HLREG)          ;SET HL TO ORIGINAL VALUE BEFORE CAL
C9         RET                      ; SUBROUTINE

```


XX DUAD-PC ASSEMBLER (A.C.P.) XX

PAGE 34

```

;XXXXXXXXXXXXXXXXXXXX END SUBROUTINE SRET XXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXX START SUBROUTINE OF COMSTR XXXXXXXXXXXXXXXXXXXXXXX
3E00 COMSTR:LD A,0
321070 LD (STRFLG),A ;SET FLAG FIRST CHARECTER DR NOT
1A COMPS1:LD A,(DE)
BE CP (HL)
C25C6F JP NZ,COMPS2
13 INC DE
23 INC HL
3E01 LD A,01H
321070 LD (STRFLG),A
C34D6F JP COMPS1
FE3A COMPS2:CP 03AH ;COMP (DE) = :
C2696F JP NZ,COMPS3
3A1070 LD A,(STRFLG)
FE01 CP 01H
CA6F6F JP Z,STROK
3EFF COMPS3:LD A,0FFH
2A0A70 LD HL,(STALB) ;SET HL TO POINT TO FIRST CHAR. OG L
C9 RET
ED530C70 STROK: LD (CURLN),DE
3A856E LD A,(CHKMLB)
3C INC A
32856E LD (CHKMLB),A
32876E LD (FLGLB),A
3E00 LD A,0H ;SET FLAG TO MAIN PROGRAM THAT COMP OK
C9 RET
;XXXXXXXXXXXXXXXXXXXX END SUBROUTINE OF COMPSTR XXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXX START SUBROUTINE SAVENT XXXXXXXXXXXXXXXXXXXXXXX
D5 SAVENT: PUSH DE
1A LD A,(DE)
320E70 LD (ADDNL),A ;
13 INC DE ;
1A LD A,(DE) ; SAVE ADD OF NEXT LINE IN ADDNL
320F70 LD (ADDNL+1),A ;
D1 POP DE
C9 RET
;XXXXXXXXXXXXXXXXXXXX END SUBROUTINE SEVENT XXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXX START SUBROUTINE DESKBI XXXXXXXXXXXXXXXXXXXXXXX
; DE SKIP BLANK BY INCREMENT DE
1A DESKBI:LD A,(DE)
FE20 CP 20H
C2966F JP NZ,DESB11
13 INC DE
C38C6F JP DESKBI
C9 DESB11:RET
;XXXXXXXXXXXXXXXXXXXX END SUBROUTINE DESKBI XXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXX START SUBROUTINE HLSKBI XXXXXXXXXXXXXXXXXXXXXXX
; HL SKIP BLANK BY INCREMENT HL
7E HLSKBI:LD A,(HL)
FE20 CP 20H
C2A16F JP NZ,HLSB11
23 INC HL
C3976F JP HLSKBI
C9 HLSB11:RET
;XXXXXXXXXXXXXXXXXXXX END SUBROUTINE HLSKBI XXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXX START SUBROUTINE NEXTLG XXXXXXXXXXXXXXXXXXXXXXX
ED5B0E70 NEXTLG:LD DE,(ADDNL)
C0806F CALL SAVENT ;SAVE ADD OF NEXT LINE
C0B76F CALL CHKEOF ;CHECK END OF TEXT
13 INC DE

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 35

```

13          INC DE
13          INC DE
13          INC DE
CD8C6F      CALL DESKBI
2A8A78      LD HL,(STALB) ;SET HL TO POINT TO FIRST CHAR. OF L
C9          RET ;GOSUBL STATEMENT
;XXXXXXXXXXXXXXXXXXXX END SUBROUTINE NEXTLG XXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXX START SUBROUTINE CHKEOF XXXXXXXXXXXXXXXXXXXXXXX
13          CHKEOF:INC DE
1A          LD A,(DE)
FE00        CP 00H
C2C46F      JP NZ,ECKEOF
;+++++ CALL ERROR +++++
3E00        LD A,0
32866E      LD (EOFTXT),A
C9          RET
;+++++ END ERROR +++++
1B          ECKEOF:DEC DE
C9          RET
;XXXXXXXXXXXXXXXXXXXX END SUBROUTINE CHKEOF XXXXXXXXXXXXXXXXXXXXXXX
;***** END GOSUB LABEL COMMAND *****
|-----|
|          RETURN LABEL COMMAND          |
|-----|
;XXXXXXXXXXXXXXXXXXXX START RETURN LABEL STATEMENT XXXXXXXXXXXXXXXXXXXXXXX
|-----|
C0          RET NZ
;----- END CHECK SYNTAX ERROR -----
;+++++ SET TEMPORY STACK POINTER +++++
F5          PUSH AF
3A5073      LD A,(STKPNT)
D608        SUB 8
325073      LD (STKPNT),A
F1          POP AF
;+++++ END SET TEMPORY STACK POINTER +++++
ED530878   RETLAB:LD (GR7),DE
D1          POP DE ;POP RETURN TO SYSTEM ADDRESS
ED536478   LD (RETSYS),DE
F1          POP AF ;POP TOKEN OF GOSUBL
FEB6        CP 006H ;COMP WITH TOKEN OF GOSUBL
1E2B        LD E,20H ;DISPLAY "RETURN LABEL WITHOUT GOSUB
C2F93B      JP NZ,3BF9H ;JUMP TO DISP ERROR AND RET TO BASIC
D1          POP DE ;POP EXECLN
ED5352E8   LD (0EB52H),DE ;CHANGE EXECLN TO NEW VALUE
E1          POP HL ;HL = HL1 POINT TO HEADER OF LINE
D1          POP DE ;POP ARYEND
ED538267   LD (ARYEND),DE ;POP ARYBGN
D1          POP DE ;POP VAREND
ED538067   LD (ARYBGN),DE ;POP VAREND
D1          POP DE ;POP VARBGN
ED537E67   LD (VAREND),DE ;POP VARBGN
D1          POP DE ;POP VARBGN
ED537C67   LD (VARBGN),DE ;POP VARBGN
ED5B6470   LD DE,(RETSYS)
D5          PUSH DE ;PUSH RETURN TO SYSTEM ADDRESS
ED5B0870   LD DE,(GR7)
C9          RET
;***** END RETURN LABEL STATEMENT *****
;XXXXXXXXXXXXXXXXXXXX DEFINE STORAGE XXXXXXXXXXXXXXXXXXXXXXX

```



```

0000 GR6: DW 0
0000 GR7: DW 0
0000 STALB: DW 0
0000 CURLN: DW 0
;XXXXXXXXXXXXXXXXXXXX HLREG: DW 0
0000 ADDNL: DW 0
00 STRFLG:DB 0
;
;***** END GOSUB LABEL COMMAND *****
;
;
;-----
;
;-----
;
ED430670 RETURN:LD (GR6),BC ;SAVE BC IN GR6
;----- SET TEMPORY STACK POINTER -----
F5 PUSH AF
3A5073 LD A,(STKPNT)
D608 SUB 8
325073 LD (STKPNT),A
F1 POP AF
;----- END SET TEMPORY STACK POINTER -----
C1 POP BC ;POP RET TO SYSTEM ADDRESS
ED436470 LD (RETSYS),BC ;SAVE RET TO SYSTEM ADD IN RETSYS
3B DEC SP ;TO COMPENSATE INC SP IN MAIN PROGRA
C1 POP BC ;POP AF FLAG
ED436670 LD (AFREG),BC ;SAVE AF REG IN AFREG
C1 POP BC ;POP EXECLN
ED436870 LD (EXECLN),BC ;SAVE EXECLN IN EXECLN
C1 POP BC ;POP HL POINT TO HEADER OF LINE
ED436A70 LD (HL1),BC ;SAVE HL IN HL1
;***** RETURN PREVIOUS LOCAL AREA *****
C1 POP BC ;POP ARYEND IN BC
ED438267 LD (ARYEND),BC ;SET ARYEND TO PREVIOUS LOCAL AREA
C1 POP BC ;POP ARYBGN IN BC
ED430067 LD (ARYBGN),BC ;SET ARYBGN TO PREVIOUS LOCAL AREA
C1 POP BC ;POP VAREND IN BC
ED437E67 LD (VAREND),BC ;SET VAREND TO PREVIOUS LOCAL AREA
C1 POP BC ;POP VARBGN IN BC
ED437C67 LD (VARBGN),BC ;SET VARBGN TO PREVIOUS LOCAL AREA
;***** END RETURN PREVIOUS LOCAL AREA *****
ED4B6A70 LD BC,(HL1) ;+
C5 PUSH BC ;+
ED4B6870 LD BC,(EXECLN) ;+
C5 PUSH BC ;+
ED4B6670 LD BC,(AFREG) ;+SET STACK TO THE SAME CONDITION
C5 PUSH BC ;+ BEFORE GO TO THIS ROTINE
33 INC SP ;+COMPENSATE DEC SP
ED4B6470 LD BC,(RETSYS) ;+
C5 PUSH BC ;+
ED4B0670 LD BC,(GR6) ;SET BC TO INITIAL VALUE
C3A345 JP 045A3H ;JP TO OLD PROGRAM IN SYSTEM
;***** DEFINE STORAGE *****
0000 RETSYS:DW 0
0000 AFREG: DW 0
0000 EXECLN:DW 0
0000 HL1: DW 0
0000 DE1: DW 0
;***** END RETURN COMMAND *****
;
;

```


GOSUB COMMAND

```

-----
ED430670 GOSUB: LD (GR6),BC ;SAVE BC REG IN GR6
ED530870 LD (GR7),DE ;SAVE DE REG IN GR7
-----
; SET ADDRESS OF VARIABLE TABLE INTO TEMPORY STACK -----
C5 PUSH BC
DDE5 PUSH IX
F5 PUSH AF
DD2A4E73 LD IX,(BASESK)
3A5073 LD A,(STKPNT)
4F LD C,A
0600 LD B,0
DD09 ADD IX,BC
ED4B7C67 LD BC,(VARBGN)
DD7100 LD (IX),C
DD23 INC IX
DD7000 LD (IX),B
DD23 INC IX
ED4B7E67 LD BC,(VAREND)
DD7100 LD (IX),C
DD23 INC IX
DD7000 LD (IX),B
DD23 INC IX
ED4B8067 LD BC,(ARYBGN)
DD7100 LD (IX),C
DD23 INC IX
DD7000 LD (IX),B
DD23 INC IX
ED4B8267 LD BC,(ARYEND)
DD7100 LD (IX),C
DD23 INC IX
DD7000 LD (IX),B
C600 ADD A,B
325073 LD (STKPNT),A
F1 POP AF
DDE1 POP IX
C1 POP BC
;-----
; END SET ADDRESS OF VARIABLE TABLE INTO TEMPORY STACK --
C1 POP BC ;POP RET TO SYSTEM ADDRESS
ED5B7C67 LD DE,(VARBGN) ;+
D5 PUSH DE ;+
ED5B7E67 LD DE,(VAREND) ;+
D5 PUSH DE ;+ PUSH PREVIOUS LOCAL AREA IN STACK
ED5B8067 LD DE,(ARYBGN) ;+
D5 PUSH DE ;+
ED5B8267 LD DE,(ARYEND) ;+
D5 PUSH DE ;+
ED537C67 LD (VARBGN),DE ;X
ED537E67 LD (VAREND),DE ;X SET FOR NEW LOCAL AREA
ED538067 LD (ARYBGN),DE ;X
C5 PUSH BC ;PUSH RET TO SYSTEM ADDRESS
ED5B8870 LD DE,(GR7)
ED4B0670 LD BC,(GR6)
C35545 JP 04555H ;JP TO PROGRAM IN SYSTEM
;***** END GOSUB COMMAND *****
-----

```

LABEL COMMAND

** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 38

```

23 LABEL: INC HL
7E LD A,(HL)
FE3A CP 03AH
CA0671 JP Z,LABEL2
FE00 CP 00
CA0171 JP Z,LABEL1
C3F270 JP LABEL
1E02 LABEL1:LD E,2
C3F93B JP 3BF9H
23 LABEL2:INC HL
7E LD A,(HL)
FE00 CP 00
1E02 LD E,2
C2F93B JP NZ,3BF9H
C9 RET

```

```

-----
;
; GLOBAL COMMAND
;
-----

```

```

2B GOBOL0:DEC HL
D7 RST 10H
C8 RET Z
CD9B40 CALL 409BH
2C DB ',,' ; Z00 STATEMENT INC L XXXXXXXX

```

```

;XXXXXXXXX GOBOL STATEMENT XXXXXXXXXXXXXXXX
;

```

```

011071 GOBOL: LD BC,GOBOL0
C5 PUSH BC
; XXX DB 0F6H XXX
;
; GET_VARIABLE
;

```

```

;GETVAR: REM BECAUSE IT MAY BE THE SAME NAME OF LOCAL SUBROUTINE
AF XOR A
3244EF LD (0EF44H),A
46 LD B,(HL)
;GTVAR: REM BECAUSE IT MAY BE THE SAME NAME OF LOCAL SUBROUTINE
CD9844 CALL 4498H
DADF3B JP C,3BDFH
C34A4E JP 4E4AH

```

```

;***** END GLOBAL COMMAND *****
;

```

```

;
; GET LOCAL VARIABLE
;
-----

```

```

;GET LOCAL VARIABLE
;
;----- CHECK LOCAL SWITCH ON/OFF -----
3A5173 LD A,(LCLON)
FE00 CP 0 ; LCLON = 0 ---> SWITCH OFF
C23271 JP NZ,GETLV1 ; USE TO USE LOCAL COMMAND GO TO GET
C9 RET ; NEVER USE LOCAL COMMAND RET TO GLOBAL
;----- END CHECK LOCAL SWITCH ON/OFF -----

```

```

ED430670 GETLV1:LD (GR6),BC
ED4B7C67 LD BC,(VARBN)
;

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 39

```

ED435373      LD      (TMPVB),BC      ;
ED487E67      LD      BC,(VAREND)      ;
ED435373      LD      (TMPVE),BC      ;
ED488867      LD      BC,(ARYBGN)      ; SAVE CURRENT LOCAL AREA VARIABLE TA
ED435773      LD      (TMPAB),BC      ; USE FOR NEXT VARIABLE
ED488267      LD      BC,(ARYEND)      ;
ED435973      LD      (TMPAE),BC      ;

016A71        LD      BC,GETLV2        ;*
C5            PUSH   BC                ;* SET WHEN RET TO COME BACK TO GETL
3A5873        LD      A,(STKPNT)        ;
FE00         CP      0                ;
CA7671        JP      Z,GETLV3         ;
D600         SUB   0                ;*
325273        LD      (TMPSPK),A        ;
C3D571        JP      ENDLV4           ;* FOR THE CURRENT LOCAL VAR

016A71        GETLV2:LD      BC,GETLV2   ; LOAD ADDRESS OF GETLV2. ----> BC
C5            PUSH   BC                ; WHEN RET WILL COME BACK TO GETLV2
3A5273        LD      A,(TMPSPK)        ;
FE00         CP      0                ; CHECK LEVEL OF GOSUB
C29C71        JP      NZ,GETLV4        ; JUMP IF IS NOT THE FIRST LEVEL OF

ED485373      GETLV3:LD      BC,(TMPVB)  ;
ED437C67      LD      (VARBGN),BC      ;
ED485573      LD      BC,(TMPVE)      ;
ED437E67      LD      (VAREND),BC      ; SET VAR TO CURRENT LOCAL VALUE BE
ED485773      LD      BC,(TMPAB)      ;
ED438867      LD      (ARYBGN),BC      ;
ED485973      LD      BC,(TMPAE)      ;
ED438267      LD      (ARYEND),BC      ;

ED488670      LD      BC,(GR6)         ;
F1            POP   AF                ; POP TO ELIMINATE GETLV2 ADD
;            ;***** TEST POINT A *****
;            PUSH   AF
;            LD      A,41H
;            CALL   0257H
;            POP   AF
;            ;***** END TEST POINT A *****
C9            RET

GETLV4:
DD2A4E73      LD      IX,(BASESK)       ; CONTENT OF BASESK IS START ADDRESS
4F           LD      C,A               ; DISPLACE ----> C REG
0600         LD      B,0               ; CLEAR B = 0 FOR USE TO ADD
DD09         ADD   IX,BC               ; IX = BASE ADDRESS + DISPLACEMENT

DD4E00         LD      C,(IX)          ;
DD4601         LD      B,(IX+1)        ;
ED437C67      LD      (VARBGN),BC      ;
DD4E02         LD      C,(IX+2)        ;
DD4603         LD      B,(IX+3)        ;
ED437E67      LD      (VAREND),BC      ; SET VARIABLE TABLE TO SEARCH IN TH
DD4E04         LD      C,(IX+4)        ;
DD4605         LD      B,(IX+5)        ;
ED438867      LD      (ARYBGN),BC      ;
DD4E06         LD      C,(IX+6)        ;
DD4607         LD      B,(IX+7)        ;
ED438267      LD      (ARYEND),BC      ;

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 40

```

3A5273      LD  A,(TMPSKP)      ;*
D48B        SUB  8              ;* SUB FOR END OF NEXT HIGHER LEVEL
325273      LD  (TMPSKP),A      ;*

```

```

|||||***** TEST POINT B *****
;          PUSH  AF
;          LD   A,42H
;          CALL 0257H
;          POP  AF
|||||***** END TEST POINT B *****

```

```
ED4B0670 ENDLV4:LD  BC,(GR6)
```

```

F5      GTLVR: PUSH AF          ;GTLVR =GET LOCAL VARIABLE
C5      PUSH BC
D5      PUSH DE
E5      PUSH HL
AF      XOR  A
4F      LD  C,A
D7      RST 10H
3885    JR  C,GTVR2
CD9944  CALL 4499H
3889    JR  C,GTVR4
4F      GTVR2: LD  C,A
D7      GTVR3: RST 10H
38FD    JR  C,GTVR3
CD9944  CALL 4499H
38F8    JR  NC,GTVR3
111772  GTVR4: LD  DE,GTVR5
D5      PUSH DE
1602    LD  D,2
FE25    CP  '%'
C8      RET  Z
14      INC  D
FE24    CP  '0'
C8      RET  Z
14      INC  D
FE21    CP  '!'
C8      RET  Z
1608    LD  D,8
FE23    CP  'N'
C8      RET  Z
78      LD  A,B
D641    SUB  'A'
E67F    AND  7FH
5F      LD  E,A
1600    LD  D,0
E5      PUSH HL
21A8EF  LD  HL,8EFA8H
19      ADD  HL,DE
56      LD  D,(HL)
E1      POP  HL
2B      DEC  HL
C9      RET
7A      GTVR5: LD  A,D
3245EF  LD  (0EF45H),A
D7      RST 10H
3A83EF  LD  A,(0EF83H)
3D      DEC  A
CAC372  JP  Z,GTVR4

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

XX DUAD-PC ASSEMBLER (A.C.P.) XX

```

F23172      JP  P,GTVR6
7E          LD  A,(HL)
D628       SUB  ''
CA9772     JP  Z,GETRY
D633       SUB  33H
CA9772     JP  Z,GETRY
AF          GTVR6: XOR  A
3283EF     LD  (0EF83H),A
E5         PUSH HL
D5         PUSH DE
3A95F8     LD  A,(0F095H)
B7         OR  A
3292F8     LD  (0F092H),A
CA7A72     JP  Z,GTURA
2AC4EF     LD  HL,(0EFC4H)
11C6EF     LD  DE,0EFC6H
19         ADD  HL,DE
227E67     LD  (VAREND),HL      ;VAREND *****
EB         EX  DE,HL
EB          GTVR7: EX  DE,HL
2A7E67     LD  HL,(VAREND)      ;VAREND
CD9548     CALL 4895H
281A       JR  Z,GTVR9
E1         POP  HL
1A         LD  A,(DE)
6F         LD  L,A
BC         CP  H
13         INC  DE
280B       JR  NZ,GTVR8
1A         LD  A,(DE)
B9         CP  C
2807       JR  NZ,GTVR8
13         INC  DE
1A         LD  A,(DE)
B8         CP  B
CABD72     JP  Z,NEWV2
3E         DB  3EH
13          GTVR8: INC  DE
13          INC  DE
E5         PUSH HL
2600       LD  H,0
19         ADD  HL,DE
18DD       JR  GTVR7
3A92F8     GTVR9: LD  A,(0F092H)
B7         OR  A
CAB672     JP  Z,NEWVR
AF         XOR  A
3292F8     LD  (0F092H),A
2A8067     GTVRA: LD  HL,(ARYBGN)      ;ARYBGN
227E67     LD  (VAREND),HL      ;VAREND
2A7C67     LD  HL,(VARBGN)      ;VARBGN
C34C72     JP  GTVR7
;
;CREATE NEW VARIABLE
;
E1          NEWVR: POP  HL      ;POP TO SET INTERIOR PUSH
E1          POP  HL      ;POP TO SET INTERIOR PUSH
E1          POP  HL      ;POP TO SET TO BE IN THE SAME STATE
D1          POP  DE      ;POP TO SET TO BE IN THE SAME STATE
C1          POP  BC      ;POP TO SET TO BE IN THE SAME STATE
F1          POP  AF
    
```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 42

```

C9          RET
13          NEW2: INC DE
E1          POP HL
C1          POP BC          ;ELIMINATE ST TO SET RETURN
C1          POP BC          ;SET TO BE IN THE SAME STATE
C1          POP BC
C1          POP BC
C37671     JP   GETLV3      ; RET

;
;GET ARRAY  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;
E5          GTRY1: PUSH HL
2A44EF     LD   HL,(0EF44H)
E3          EX  (SP),HL
57          LD  D,A
D5          GTRY1: PUSH DE
C5          PUSH BC
CDA044     CALL 44A0H
C1          POP  BC
F1          POP  AF
EB          EX  DE,HL
E3          EX  (SP),HL
E5          PUSH HL
EB          EX  DE,HL
3C          INC  A
57          LD  D,A
7E          LD  A,(HL)
FE2C       CP   ' '
20EE       JR   Z,GTRY1
FE5D       CP   ' '
2003       JR   NZ,GTRY2
D7         RST  10H
1804       JR   GTRY3

;
CD9B40     GTRY2: CALL 409BH
29         DB   ' ) '

;
229AEF     GTRY3: LD   (0EF9AH),HL
E1         POP  HL
2244EF     LD   (0EF44H),HL
1E00       LD   E,0
D5         PUSH DE
11         DB   11H
E5         GTRY4: PUSH HL
F5         PUSH AF
2A0067     LD   HL,(ARYBGN) ;ARYBGN XXXXXXXXXXXXXXXXXXXXXXX
3E         DB   3EH

;
19         GTRY5: ADD  HL,DE
EB         EX   DE,HL
2A0267     LD   HL,(ARYEND) ;ARYEND XXXXXXXXXXXXXXXXXXXXXXX
EB         EX   DE,HL
CD9540     CALL 4095H
3A45EF     LD   A,(0EF45H)
2031       JR   Z,MAKRY
BE         CP   (HL)
23         INC  HL
2008       JR   NZ,GTRY6
7E         LD   A,(HL)
09         CP   C

```


** DUAD-PC ASSEMBLER (A.C.P.) **

PAGE 44

```

19          ADD HL,DE
F1          POP AF
3D          DEC A
44          LD B,H
4D          LD C,L
28E9       JR NZ,GTRY9
3A45EF     LD A,(0EF45H)
44          LD B,H
4D          LD C,L
29          ADD HL,HL
D604       SUB 4H
3804       JR C,GTRYA
29          ADD HL,HL
2806       JR Z,GTRYB
29          ADD HL,HL
B7         GTRYA: OR A
E24073     JP PD,GTRYB
09         ADD HL,BC

;
C1         GTRYB: POP BC
09         ADD HL,BC
EB         EX DE,HL
2A9AEF     GTRYC: LD HL,(0EF9AH)
F1         POP AF
F1         POP AF
F1         POP AF
F1         POP AF
F1         POP AF
C37671     JP GETLV3          ; RET
;
;----- DECLARE VARIABLE-----
;
007D       BASESK:DW 7D00H          ;START ADDRESS OF TEMPORARY STACK
00         STKPNT:DB 0H            ;STACK POINTER FOR TEMPORARY STACK
00         LCLON:DB 0H            ;FLAG SWITH ON/OFF TO INDICATE LOCAL IS
00         TMPSKP:DB 0H          ;TEMPORARY STACK POINTER USE TO SEARCH
0000       TMPVB:DW 0H            ;TEMPORARY VAR TO SAVE VARGBN
0000       TMPVE:DW 0H            ;TEMPORARY VAR TO SAVE VAREND
0000       TMPAB:DW 0H            ;TEMPORARY VAR TO SAVE ARYBGN
0000       TMPAE:DW 0H            ;TEMPORARY VAR TO SAVE ARYEND

;***** END GET LOCAL VARIABLE *****
;-----
;
; DIM COMMAND
;-----
;
2B         DIM0: DEC HL
D7         RST 10H
C8         RET 2
CD9B40     CALL 409BH
2C         DB ','                ; 280 STATEMENT INC L *****

;***** DIM STATEMENT *****
;
015873     DIM: LD BC,DIM0
C5         PUSH BC
F6         DB 0F6H

;
; GET VARIABLE
;

```


ภาคผนวก ข

ขั้นตอนการทำงานของคำสั่ง (ALGORITHM)

เป็นการแสดงขั้นตอนการทำงานของคำสั่งต่าง ๆ ที่เพิ่มเข้าไป ส่วนตัวโปรแกรมจริง ๆ นั้นปรากฏอยู่ในภาคผนวก

```

IFS : PROCEDURE
CHECK SYNTAX (*ทำหน้าที่ในการตรวจสอบความถูกต้องรูปแบบของคำสั่ง*)
CALL FINDADDRESS (* ทำหน้าที่หาคำแหน่งของ THENDO, ELSEDO และ
ENDIF ที่คู่กัน*)
CALL TEST CONDITION (*เป็นการตรวจสอบเงื่อนไขที่กำหนด เป็นการเรียกใช้หน่วย
ของโปรแกรมในอินเทอร์พรีเตอร์*)
POP RET-ADDRESS
PUSH END-ADDRESS
PUSH LINE-NUMBER OF END-STATEMENT
PUSH TOKEN OF IFS (*เก็บไว้เพื่อตรวจสอบคำสั่ง THENDO, ELSEDO,
ENDIF ว่าต้องมีคำสั่ง IFS: ที่คู่กัน *)
PUSH RET-ADDRESS
IF CONDITION - TRUE
THEN SET POINTER TO THEN-STATEMENT
ELSE SET POINTER TO ELSE-STATEMENT
ENDIF
RET
END : PROC

```

รูปที่ข. 1 แสดงขั้นตอนการทำงานของคำสั่ง IFS

FINDADDRESS : PROCEDURE

SET FLAG -X=Y=Z=0=FALSE (*แฟล็กควบคุมการค้นหา THEN-ELSE-ENDIF ครบชุดหรือยัง*)
 - FLGTHEN = FLGELSE = FLGNIF = 1 (*แฟล็กควบคุมการ NESTED*)
 - FLGSYN = 3 (*แฟล็กควบคุม sequence ของ IF-THEN-ELSE-ENDIF*)

DO WHILE X AND Y AND Z=FALSE

GET NEXT LINE

IF END OF TEXT

THEN IF X = 0 THEN DISPLAY "IFS WITHOUT THENDO"

IF Y = 0 THEN DISPLAY "IFS WITHOUT ELSEDO"

IF Z = 0 THEN DISPLAY "IFS WITHOUT ENDIF"

ENDIF

GET TOKEN

IF TOKEN = "IFS"

THEN INC FLGTHEN, FLGELSE, FLGNIF (*ถ้าเป็นคำสั่ง IFS เพิ่มค่าให้กับ
แฟล็กของ THEN ของ ELSE ของ IF อย่างละ 1*)

ENDIF

IF TOKEN = "THENDO"

THEN DO

DEC FLGTHEN (*ลดค่าแฟล็กของ THEN*)

IF FLGTHEN = 0

THEN DO พบ ADDRESS ของ THEN ที่คู่กัน

THENADD = ADDRESS-OF-CURRENT-LINE

IF FLGSYN = 3

THEN DEC FLGSYN(*ลดค่าแฟล็กของการตรวจสอบลำดับ*)

ELSE DISPLAY "MIS-SEQUENCE"

ENDIF

รูปที่ ข. 1 (ต่อ) แสดงขั้นตอนการทำงานของคำสั่ง IFS


```

X = TRUE = 1 (บอกให้รู้ว่าเจอ THEN แล้ว)
    ENDDO
ENDIF

ENDO

ENDIF

IF TOKEN = "ELSEDO"

    THEN DO

        DEC FLGELSE

        IF FLGELSE = 0

            THEN DO พบ ADDRESS ของ ELSE ที่ถูกค้น

                ELSEADD = ADDRESS-OF-CURRENT-LINE
                Y = TRUE = 1 (บอกให้รู้ว่าพบ ELSE แล้ว)
                IF FLGSYN = 2

                    THEN DEC FLGSYN

                    ELSE DISPLAY "MIS-SEQUENCE"

                ENDIF

            ENDDO

        ENDIF

    ENDDO

ENDIF

ENDDO

ENDIF

IF TOKEN = "ENDIF"

    THEN DO

        DEC FLGNIF

        IF FLGNIF = 0

            THEN DO

```

รูปที่ ๑. ๑ (ต่อ) แสดงขั้นตอนการทำงานของคำสั่ง IFS


```

ENDIFADD = ADDRESS-OF-CURRENT-LINE
Z = TRUE
IF FLGSYN = 1
    THEN DEC FLGSYN
    ELSE DISPLAY "MIS-SEQUENCE"
ENDIF
ENDDO
ENDIF
ENDDO
ENDIF
END EHILE
END : PROC

```

รูปที่ ข. 1 (ต่อ) แสดงขั้นตอนการทำงานของคำสั่ง IFS

```

THENDO : PROCEDURE
CHECK SYNTAX
POP RET-ADDRESS   เอาค่าแอสแอดเรสของ RET-ADDRESS   ไปเก็บ
POP TOKEN
IF TOKEN ≠ "IFS"
    THEN DISPLAY "THENDO WITHOUT IFS"
ENDIF
PUSH TOKEN OF THENDO
PUSH RET-ADDRESS   เอาค่าแอสแอดเรสของ RET-ADDRESS   ใสลงใน STACK
RET
END : PROC

```

รูปที่ ข. 2 แสดงขั้นตอนการทำงานของคำสั่ง THENDO


```

ELSEDO : PROCEDURE
CHECK SYNTAX
POP RET-ADDRESS
POP TOKEN
IF TOKEN ≠ "IFS"
    THEN DISPLAY "ELSEDO WITHOUT IFS"
ENDIF
PUSH TOKEN OF ELSEDO
PUSH RET-ADDRESS
RET
END : PROC

```

รูปที่ ๓. ๓ แสดงขั้นตอนการทำงานของคำสั่ง ELSEDO

```

ENDDO : PROCEDURE
CHECK SYNTAX
POP TOKEN
IF TOKEN = "THENDO" .OR. TOKEN = "ELSEDO"
    THEN
        ELSE DISPLAY "ENDDO WITHOUT THENDO OR ELSEDO"
ENDIF
SET POINTER TO POINT TO ENDIF STATEMENT
PUSH RET-ADDRESS
RET
END : PROC

```

รูปที่ ๓. ๔ แสดงขั้นตอนการทำงานของคำสั่ง ENDDO


```

GOSUBL : PROCEDURE
CHECK SYNTAX
GET START OF TEXT
DO WHILE .NOT. EOF
  GET TOKEN
  IF TOKEN = "LABEL"
    THEN COMPARE LABEL NAME
      IF COMPARE OK
        THEN IF FIGLBL = 1
          THEN DISPLAY "MULTI-DEFINED LABEL NAME"
        ENDIF
        FIGLBL = 1
      ENDIF
    ENDIF
  ENDIF
  GET NEXT LINE
  IF END OF TEXT
    THEN
      IF FIGLBL  $\neq$  1
        THEN
          DISPLAY "UNDEFINED LABEL NAME"
        ELSE
          ENDIF
        ELSE
          ENDIF
      ENDIF
    ENDIF
  ENDDO

```



SET NEW AREA FOR LOCAL VARIABLE

TRANSFER CONTROL TO LINE NUMBER WHICH HAVE THE SAME LABEL NAME

END : PROC

รูปที่ ๕ (ต่อ) แสดงขั้นตอนการทำงานของคำสั่ง GOSUBL

RETURNL : PROCEDURE

CHECK SYNTAX

POP RET-ADDRESS

PCP TOKEN

IF TOKEN \neq "GOSUBL"

THEN DISPLAY "RETURNL WITHOUT GOSUBL"

ENDIF

SET POINTER TO POINT TO STATEMENT WHICH CALL THIS SUBROUTINE

CLEAR AREA OF LOCAL VARIABLE FOR THIS SUBROUTINE

PUSH RET-ADDRESS

RET

END : PROC

รูปที่ ๖ แสดงขั้นตอนการทำงานของคำสั่ง RETURNL


```

LOOP : PROCEDURE

CHECK SYNTAX

CALL FIND ENDLOOP

POP RET-ADDRESS

PUSH ADDRESS OF STATEMENT AFTER ENDLOOP

PUSH LINE NUMBER OF STATEMENT AFTER ENDLOOP

PUSH LOOP-ADDRESS

PUSH LINE-NUMBER OF LOOP STATEMENT

PUSH TOKEN OF LOOP

PUSH RET-ADDRESS

RET

END. : PROC

```

รูปที่ ๑. 7 แสดงขั้นตอนการทำงานของคำสั่ง LOOP

```

FINENDLOOP : PROCEDURE

SET FLAG = 1

X = 1

DO WHILE X = FALSE = 0

GET NEXT LINE

IF END OF TEXT

THEN DO

DISPLAY "LOOP WITHOUT ENDLOOP"

X = TRUE = 1

ENDDO

ENDIF

```

รูปที่ ๑.8


```

GET TOKEN

IF TOKEN = "LOOP" THEN INC FLGLP

IF TOKEN = "END LOOP"
  THEN DO
    DEC FLGLP
    IF FLGLP = 0
      THEN DO
        ENDLLOOP-ADDRESS = THAT LINE ADDRESS
        X = TRUE = 1
      ENDDO
    ENDIF
  ENDDO
ENDIF

ENDDO

END : PROC

```

รูปที่ ๗. ๘ (ต่อ)

```

EXIT WHEN : PROCEDURE

CHECK SYNTAX

TEST CONDITION

IF CONDITION = TRUE
  THEN DO
    POP RET ADDRESS
    POP TOKEN

    IF TOKEN ≠ "LOOP"
      THEN DISPLAY "EXIT WHEN WITHOUT LOOP"

```

รูปที่ ๗. ๙


```

        ENDIF
        SET POINTER TO POINT TO STATEMENT AFTER ENDLOOP STATEMENT
    ENDDO
ENDIF
RET
END : PROC

```

รูปที่ ๙.๙ (ต่อ) แสดงขั้นตอนการทำงานของคำสั่ง EXIT WHEN

```

ENDLOOP : PROCEDURE
    CHECK SYNTAX
    POP RET ADDRESS
    POP TOKEN
    IF TOKEN ≠ "LOOP"
        THEN DISPLAY "ENDLOOP WITHOUT LOOP"
    ENDIF
    SET POINTER TO POINT TO STATEMENT AFTER LOOP STATEMENT
    PUSH INFORMATION OF LOOP IN STACK (FOR NEXT TIME OF ENDLOOP)
    RET
END : PROC

```

รูปที่ ๙.๑๐ แสดงขั้นตอนการทำงานของคำสั่ง ENDLOOP

ภาคผนวก ค.

โปรแกรมที่ใช้ในการทดสอบอินเตอร์พรีเตอร์ใหม่

1. ทดสอบคำสั่งใหม่

ก ทดสอบกลุ่มคำสั่ง IFS

```

10 LPRINT "TEST IF PROGRAM"
20 A=10
30 IFS A=20
40   THENDO
50     LPRINT "11111111111111111111"
60     LPRINT "22222222222222222222"
70     LPRINT "33333333333333333333"
80     LPRINT "Then Ok"
90     E=321
100    F=100
110    G=EXF
120    LPRINT "E = ";E,"F = ";F,"G = ";G
130  ENDDO
140  ELSEDO
150    LPRINT "AAAAAAAAAAAAAAAAAAAA"
160    LPRINT "BBBBBBBBBBBBBBBBBBBB"
170    LPRINT "CCCCCCCCCCCCCCCC"
180    LPRINT "Else Ok"
190    B = 123
200    C = 200
210    D = BxC
220    LPRINT "B = ";B,"C = ";C,"D = ",D
230  IFS B=123
240    THENDO
250      LPRINT "Second If Then Level 2"
260    ENDDO
270    ELSEDO
280      LPRINT "Second If Else Level 2"
290    ENDDO
300  ENDIF
310  ENDDO
320 ENDIF

```

```

TEST IF PROGRAM
AAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBB
CCCCCCCCCCCCCCCC
Else Ok
B = 123      C = 200      D =      24600
Second If Then Level 2

```


๗. ทดสอบคอมไพล์คำสั่ง LOOP

```

10 LPRINT*                PROGRAM TEST LOOP & ENLOOP COMMAND *
20 I = 1
30 LOOP
40   I=I+1
50   LPRINT "INLOOP LEVEL 1 LINE NO. 1 "
60   A=1
70   LOOP
80     LPRINT "                INLOOP ELVEL 2 LINE NO. 1 "
90     EXIT WHEN A = 3
100    LPRINT "                inloop level 2 line No. 2 "
110    A=A+1
120    ENLOOP
130    EXIT WHEN I=3
140    LPRINT "inloop level 1 line No. 2"
150 ENLOOP

```

```

PROGRAM TEST LOOP & ENLOOP COMMAND
INLOOP LEVEL 1 LINE NO. 1
INLOOP ELVEL 2 LINE NO. 1
inloop level 2 line No. 2
INLOOP ELVEL 2 LINE NO. 1
inloop level 2 line No. 2
INLOOP ELVEL 2 LINE NO. 1
inloop level 1 line No. 2
INLOOP LEVEL 1 LINE NO. 1
INLOOP ELVEL 2 LINE NO. 1
inloop level 2 line No. 2
INLOOP ELVEL 2 LINE NO. 1
inloop level 2 line No. 2
INLOOP ELVEL 2 LINE NO. 1

```

ศูนย์วิทยุทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ก. ทดสอบคำสั่ง LOCAL / GLOBAL

```

5 LPRINT"                                PROGRAM TEST LOCAL VARIABLE COMMAND"
8 GLOBAL AA,BB,CC
10 AA = 100
20 BB = 200
30 DD = 100
40 LPRINT "GLOBAL BEFORE CALL GOSUB AA =";AA,"BB =";BB
50 GOSUB 120
60 LPRINT "GLOBAL AFTER CALL GOSUB AA =";AA,"BB =";BB
70 DD = 100
80 WW = DD*AA
90 LPRINT "GLOBAL WW ";WW
100 LPRINT " END PROGRAM"
110 END
120 LOCAL AA,BB,CC,WW
130 AA = 1212
140 BB = 2222
150 WW = DD*AA
160 LPRINT " LOCAL IN LEVEL 2 AA =";AA,"BB =";BB,"WW =";WW
170 GOSUBL THREE
180 LPRINT " LOCAL IN LEVEL 2 AA =";AA,"BB =";BB
190 RETURN
200 LABEL THREE:
210 LOCAL AA,BB,CC,DD,EE
220 AA = 1313
230 BB = 2323
240 CC = 3333
250 LPRINT " LOCAL LEVEL 3 AA =";AA,"BB =";BB
255 GOSUBL FOUR
270 LPRINT " LOCAL LEVEL 3 AA =";AA,"BB =";BB,"WW=";WW
280 RETURNL
300 LABEL FOUR:
310 LOCAL AA,BB
320 AA = 4
330 BB = 44
340 LPRINT " LOCAL LEVEL 4 AA =";AA
345 GOSUBL FIVE
348 LPRINT " LOCAL LEVEL 4 AA =";AA
350 RETURNL
400 LABEL FIVE:
410 LOCAL AA
420 AA = 5
430 LPRINT " LOCAL LEVEL 5 AA =";AA;" BB=";BB;"
CC=";CC
440 RETURNL

```

```

PROGRAM TEST LOCAL VARIABLE COMMAND
GLOBAL BEFORE CALL GOSUB AA = 100 BB = 200
LOCAL IN LEVEL 2 AA = 1212 BB = 2222 WW = 121200
LOCAL LEVEL 3 AA = 1313 BB = 2323
LOCAL LEVEL 4 AA = 4
LOCAL LEVEL 5 AA = 5 BB= 44 CC= 3333
LOCAL LEVEL 4 AA = 4
LOCAL LEVEL 3 AA = 1313 BB = 2323 WW= 121200
LOCAL IN LEVEL 2 AA = 1212 BB = 2222
GLOBAL AFTER CALL GOSUB AA = 100 BB = 200
GLOBAL WW 10000
END PROGRAM

```


ง. ทดสอบกลุ่มคำสั่ง LABELED SUBROUTINE CALL

```
10 LPRINT "                TEST PROGRAM GOSUB LABEL "  
20 GOTO 60  
30 LABEL AAA:  
40 LPRINT "IN GOSUB LABEL "  
50 RETURNL  
60 LPRINT "BEFORE GOSUB LABEL "  
70 GOSUBL AAA  
80 LPRINT "AFTER GOSUB LABEL "  
90 LPRINT "END PROGRAM"
```

```
                TEST PROGRAM GOSUB LABEL  
BEFORE GOSUB LABEL  
IN GOSUB LABEL  
AFTER GOSUB LABEL  
END PROGRAM
```

ศูนย์วิทยพัชวิทยากร
จุฬาลงกรณ์มหาวิทยาลัย

2. ทดสอบโปรแกรมเดิมว่ายังคงทำงานได้เหมือนเดิม

ก โปรแกรม Mogura

```

10 REM ***** Mogura *****
20 WIDTH 40,25:COLOR6:PRINT CHR$(12);"                M O G U R A"
30 COLOR2:LOCATE13,2:PRINT"s t r i k e"                STRIKE or
40 COLOR5:LOCATE 8,5:PRINT"                            To
HIT "
50 COLOR6:LOCATE13,8:PRINT"M O G U R A"
60 COLOR5:PRINT:PRINT" You have to press the key number"
70 PRINT"                correspond to the position                of"
80 COLOR6:PRINT:PRINT"                M O G U R A"
90 PRINT:PRINT:PRINT
100 COLOR7:PRINT"                Hit RETURN key to play";
110 A$=INKEY$:IF A$(<)CHR$(13) THEN 110
120 WIDTH 80,25:CONSOLE 0,25,0,1
130 PRINT CHR$(12):COLOR2,0,1
140 LOCATE 25,4:PRINT "
150 COLOR5,0,0:LOCATE25,4:PRINT "M O G U R A  S T R I K E"
160 DIM A$(16)
170 FOR I=0 TO 16:READ A$(I):NEXT I
180 COLOR 6,0,1
190 FOR M=0 TO 105 STEP 15
200 PUT$(15+M,35)-(30+M,50),A$,PSET:BEEP :NEXT M
210 COLOR 1,0,1:LINE(9,0)-(140,70),PSET,B
220 COLOR 5,0,1:LOCATE 24,21:PRINT "<Press SPACE key to start>"
230 A$=INKEY$:IF A$="" THEN 230
240 IF A$(<)CHR$(32) THEN 230
250 BEEP:PRINT CHR$(12)
260 COLOR2,0,1:LINE(15,52)-(136,62),PSET,BF
270 COLOR7,0,0
280 LOCATE 11,14:PRINT "1"
290 LOCATE 18,14:PRINT "2"
300 LOCATE 26,14:PRINT "3"
310 LOCATE 33,14:PRINT "4"
320 LOCATE 41,14:PRINT "5"
330 LOCATE 48,14:PRINT "6"
340 LOCATE 56,14:PRINT "7"
350 LOCATE 64,14:PRINT "8"
360 REM *****
370 S=0
380 S=100:T=60
390 COLOR 5,0,1:LINE(15,8)-(135,30),PSET,BF
400 COLOR 7,0,0:LOCATE 10,3:PRINT "TIME:"; T
410 LOCATE 55,3:PRINT "SCORE:"; S
420 IF S=0 OR T=0 THEN 710
430 COLOR 6,0,1
440 A=INT(RND(3)*10):IF A>8 THEN 440
450 X=A*15:IF X<15 THEN 440
460 FOR Q=0 TO 2
470 PUT$(X,35)-(X+15,50),A$,PSET:IF Q=0 THEN BEEP 1
480 FOR J=0 TO 13 :NEXT :BEEP 0
490 IF X=15 THEN 580
500 IF X=30 THEN 590
510 IF X=45 THEN 600
520 IF X=60 THEN 610
530 IF X=75 THEN 620
540 IF X=90 THEN 630
550 IF X=105 THEN 640
560 IF X=120 THEN 650
570 IF X=135 THEN 890
580 A$=INKEY$:IF A$=CHR$(49) THEN GOSUB 950
590 A$=INKEY$:IF A$=CHR$(50) THEN GOSUB 950
600 A$=INKEY$:IF A$=CHR$(51) THEN GOSUB 950
610 A$=INKEY$:IF A$=CHR$(52) THEN GOSUB 950
620 A$=INKEY$:IF A$=CHR$(53) THEN GOSUB 950
630 A$=INKEY$:IF A$=CHR$(54) THEN GOSUB 950
640 A$=INKEY$:IF A$=CHR$(55) THEN GOSUB 950
650 A$=INKEY$:IF A$=CHR$(56) THEN GOSUB 950
660 NEXT Q

```



```

670 PUT@(X,35)-(X+15,50),AX,PRESET
680 S=S-10:T=T-1
690 FOR L=0 TO 700 :NEXT L
700 GOTO 400
710 COLOR 4,0,0:LOCATE 27,6:PRINT " G A M E   O V E R "
720 LOCATE 32,18:PRINT"REPLAY (Y/N)"
730 COLOR 5:IF S1>S THEN 750
740 S1=S
750 FOR R=0 TO 8
760 FOR U=0 TO 30 :NEXT U:LOCATE 24,20:PRINT "          "
EP 1
770 LOCATE 27,20:PRINT "TODAY'S HIGH SCORE:"; S1:BEEP .0 :NEXT R
780 COLOR 6,0,1
790 FOR M=0 TO 105 STEP 15
800 PUT@(15+M,35)-(30+M,50),AX,PSET
810 A$=INKEY$:IF A$="" THEN 820 ELSE 840
820 NEXT M
830 LINE(135,50)-(15,35),PRESET,BF:GOTO 790
840 IF A$="y" THEN 890
850 IF A$="Y" THEN 890
860 IF A$="n" THEN RUN"GAMES"
870 IF A$="N" THEN RUN"GAMES"
880 GOTO 810
890 BEEP:LINE(15,35)-(135,50),PRESET,BF
900 LOCATE 27,6:PRINT"
910 LOCATE 31,18:PRINT"
920 LOCATE 27,20:PRINT "
930 FOR Z=0 TO 700 :NEXT Z
940 GOTO 300
950 REM xxx SUB xxx
960 COLOR 3,0,1:S=S+30
970 FOR G=0 TO 45
980 BEEP 1:FOR M=0 TO 8:NEXT M
990 BEEP 0:NEXT G
1000 RETURN
1010 DATA 256,0,4080,8184,16380,32766,29646,28686
1020 DATA 29742,32766,32766,32766,32382,32766,-1,-1,-1

```

ศูนย์วิทยุทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โปรแกรม Black Jack

```

100 REM *****
110 REM XXX XXX
120 REM XXX BLACK JACK XXX
130 REM XXX Copyright 1979 (C) XXX
140 REM XXX by XXX
150 REM XXX ASCII Laboratories XXX
160 REM XXX Y&M XXX
170 REM XXX 1979/9/16 XXX
180 REM XXX XXX
190 REM *****
200 CLEAR:PRINT CHR$(12)
210 DIM CA(51),SU(51),A$(8),B$(8)
220 A$(0)=121:B$(0)=121:MY=100
230 GOSUB 1120
240 CONSOLE ,,,1:WIDTH 40,
250 REM * Main *
260 IF SF=1 THEN GOSUB 1120
270 FOR I=0 TO 2000:NEXT
280 PRINT CHR$(12):BP=0:BD=0:PL=2:DL=2:SP=0:B1=0:B2=0:FL=2:SL=2
290 COLOR 6
300 LOCATE 0,12:PRINT USING "You have $####";MY;
310 LOCATE 10,12:INPUT "How do you bet";BT
320 IF BT>MY THEN 1180
330 WIDTH 80,
340 GOSUB 1190:CX=0:CY=7:GOSUB 1220:PC(1)=TN
350 GOSUB 1190:CX=0:CY=7:GOSUB 1220:DC(1)=TN
360 GOSUB 1190:CX=16:CY=7:GOSUB 1220:PC(2)=TN:P=TS
370 GOSUB 1190:COLOR 4,0,1:LINE (30,97)-(16,71),PSET,BF:DC(2)=TN:D=TS
380 COLOR 7:GOSUB 1340:IF TC=21 THEN LOCATE 16,5:PRINT "Blackjack!";:BP=1
390 GOSUB 1390:IF TD=21 THEN LOCATE 16,21:PRINT "BlackJack!";:BD=1:GOTO 500
400 IF BP=1 THEN 500
410 REM * Player *
420 CX=16:CY=7:IF PC(1)=PC(2) THEN IF MY>BT*2 THEN LOCATE 17,6:PRINT "or Sprit?";:SP=1
430 LOCATE CX/2+8,5:PRINT "Stand or Hit?"+CHR$(7);
440 A$=INKEY$:IF A$="s" THEN LOCATE CX/2+8,5:PRINT " ";:GOTO 500
450 IF A$="x" AND SP=1 THEN 930
460 IF A$<>"h" THEN 440
470 CX=CX+16:GOSUB 1190:GOSUB 1220:PL=PL+1:PC(PL)=TN
480 GOSUB 1340:IF TC>=22 THEN LOCATE CX/2+8,5:PRINT "Burst!";:BP=-1 ELSE 430
490 REM * Dealer *
500 COLOR 4,0,1:LINE (29,96)-(17,72),PRESET,BF
510 TS=D:TN=DC(2):CX=16:CY=7:GOSUB 1220
520 IF SP=1 THEN IF B1<>0 OR B2<>0 OR (TD)=T1 AND TD>=T2 THEN 560 ELSE 540
530 IF BP<>0 OR TD>=TC THEN 560
540 CX=CX+16:DL=DL+1:GOSUB 1190:GOSUB 1220:DC(DL)=TN
550 GOSUB 1390:IF TD>=22 THEN LOCATE CX/2+8,21:PRINT "Burst!";:BD=-1 ELSE 530
560 IF SP=1 THEN 730
570 IF BP=-1 THEN 660
580 IF BD=-1 THEN MY=MY+BT:GOTO 640
590 IF BD=1 AND BP=0 THEN 660
600 IF BP=1 THEN MY=MY+BT*1.5:GOTO 640
610 IF TC=TD THEN 650
620 IF TC<TD THEN 660
630 MY=MY+BT:GOTO 640
640 LOCATE 40,0:PRINT "You win!";:GOTO 260
650 LOCATE 40,0:PRINT "Push!";:GOTO 260
660 LOCATE 40,0:PRINT "You lose.";:MY=MY-BT:IF MY>0 THEN 260
670 LOCATE 0,0:PRINT "You bursted!!!"
680 PRINT CHR$(12):LOCATE 10,12:WIDTH 40,:COLOR 6
690 PRINT "Do you wish to replay?";
700 A$=INPUT$(1)
710 IF A$<>CHR$(110) THEN 100
720 END
730 REM XXX
740 IF B1=-1 THEN 880
750 IF B1=1 AND BD<1 THEN MY=MY+BT*1.5:GOTO 880 ELSE IF BD=1 THEN 870
760 IF BD=-1 THEN 790
770 IF TD=T1 THEN 870
780 IF TD>T1 THEN 880
790 MY=MY+BT:GOTO 860
800 IF B2=-1 THEN 910
810 IF B2=1 AND BD<1 THEN MY=MY+BT*1.5:GOTO 910 ELSE IF BD=1 THEN 900
820 IF TD=T2 THEN 900
830 IF BD=-1 THEN 850
840 IF TD>T2 THEN 910
850 MY=MY+BT:GOTO 890
860 LOCATE 40,0:PRINT "You win!";:GOTO 800
870 LOCATE 40,0:PRINT "Push";:GOTO 800
880 LOCATE 40,0:PRINT "You lose.";:MY=MY-BT:IF MY>0 THEN 800 ELSE 670
890 LOCATE 50,0:PRINT "You win!";:GOTO 250

```



```

900 LOCATE 50,0:PRINT "Push";GOTO 250
910 LOCATE 50,0:PRINT "You lose.";MY=MY-BT:IF MY>0 THEN 250 ELSE 670
920 REM * Sprit First *
930 LOCATE 16,5:PRINT " ";LOCATE 17,6:PRINT " ";
940 TN=PC(2):TS=P:CX=0:CY=39:GOSUB 1220:P2(1)=TN
950 GOSUB 1190:CX=16:CY=7:GOSUB 1220:P1(2)=TN:P1(1)=PC(1):COLOR 7
960 GOSUB 1440:IF T1=21 THEN LOCATE 16,5:PRINT "BlackJack!";B1=1:GOTO 1030
970 LOCATE CX/2+8,5:PRINT "Stay or Hit?";
980 A$=INKEY$:IF A$="s" THEN LOCATE CX/2+12,5:PRINT " ";GOTO 1030
990 IF A$("<"h" THEN 980
1000 CX=CX+16:GOSUB 1190:GOSUB 1220:FL=FL+1:P1(FL)=TN
1010 GOSUB 1440:IF T1>=22 THEN LOCATE CX/2+8,5:PRINT "Burst";B1=-1 ELSE 970
1020 REM * Sprit Second *
1030 GOSUB 1190:CX=16:CY=39:GOSUB 1220:P2(2)=TN:COLOR 7
1040 GOSUB 1490:IF T2=21 THEN LOCATE 16,13:PRINT "BlackJack!";B2=1:GOTO 1100
1050 LOCATE CX/2+8,13:PRINT "Stay or Hit?";
1060 A$=INKEY$:IF A$="s" THEN LOCATE CX/2+12,13:PRINT " ";GOTO 1100
1070 IF A$("<"h" THEN 1060
1080 CX=CX+16:GOSUB 1190:GOSUB 1220:SL=SL+1:P2(SL)=TN
1090 GOSUB 1490:IF T2>=22 THEN LOCATE CX/2+8,13:PRINT "Burst";B2=-1 ELSE 1050
1100 GOTO 500
1110 END
1120 REM ** Shuffling **
1130 CONSOLE,,0,1:COLOR 6:WIDTH 40,25
1140 LOCATE 15,12:PRINT "Shuffling-";
1150 FOR I=0 TO 51:SU(I)=1:NEXT
1160 FOR I=0 TO 51
1170 C=INT(RND(1)*52):IF SU(C)=0 THEN 1170
1180 SU(C)=0:CA(I)=C:NEXT:SF=0:RETURN
1190 REM ** Take One Card **
1200 T=CA(E):E=E+1:IF E=52 THEN E=0:SF=1
1210 TS=INT(T/13):TN=T MOD 13:RETURN
1220 REM ** Display A-Card **
1230 RESTORE
1240 FOR I=0 TO TS:FOR J=1 TO 8
1250 READ A$(J):NEXT:NEXT
1260 RESTORE 1600
1270 FOR I=0 TO TN:FOR J=1 TO 8
1280 READ B$(J):NEXT:NEXT
1290 COLOR 4,0,1:LINE (CX,CY)-(CX+14,CY+26),PSET,B
1300 IF TS=1 OR TS=2 THEN CL=2 ELSE CL=5
1310 COLOR CL,0,1:PUTE (CX+2,CY+2)-(CX+12,CY+12),A$,PSET
1320 COLOR 7,0,1:PUTE (CX+2,CY+14)-(CX+12,CY+24),B$,PSET
1330 RETURN
1340 REM ** Total of Cards for Player **
1350 TC=0:A=0:FOR I=1 TO PL:W=PC(I)+1:IF W=1 THEN A=1
1360 TC=TC-(W<=9)*W-(W=10)*10:NEXT
1370 IF A=1 THEN IF TC+10<=21 THEN TC=TC+10
1380 RETURN
1390 REM ** Total of Cards for Dealer **
1400 TD=0:A=0:FOR I=1 TO DL:W=DC(I)+1:IF W=1 THEN A=1
1410 TD=TD-(W<=9)*W-(W=10)*10:NEXT
1420 IF A=1 THEN IF TD+10<=21 THEN TD=TD+10
1430 RETURN
1440 REM ** Total of Cards for Sprit First *
1450 T1=0:A=0:FOR I=1 TO FL:W=P1(I)+1:IF W=1 THEN A=1
1460 T1=T1-(W<=9)*W-(W=10)*10:NEXT
1470 IF A=1 THEN IF T1+10<=21 THEN T1=T1+10
1480 RETURN
1490 REM ** Total of Cards for Sprit Second *
1500 T2=0:A=0:FOR I=1 TO SL:W=P2(I)+1:IF W=1 THEN A=1
1510 T2=T2-(W<=9)*W-(W=10)*10:NEXT
1520 IF A=1 THEN IF T2+10<=21 THEN T2=T2+10
1530 RETURN
1540 REM *** DATA of Cards' Marks ***
1550 DATA &h8020,&h3E03,&hE3F8,&hFFBF,&hDFFF,&h7075,&h07C0,&hFE7F:REM Spade
1560 DATA &hF10C,&hFFDE,&hFFFF,&hFFFF,&h8FFB,&hF83F,&h0380,&hFE08:REM Heart
1570 DATA &h8020,&h3E03,&hE3F8,&hFFBF,&h8FFB,&hF83F,&h0380,&hFE08:REM Diamond
1580 DATA &hC070,&h3E07,&hF7FC,&hFFFF,&hDFFF,&h7075,&h07C0,&hFE7F:REM Club
1590 REM *** DATA of Cards' Numbers ***
1600 DATA &h8020,&h2202,&h2208,&h00A0,&hF006,&h01FF,&h600C,&hFF00:REM Ace
1610 DATA &h11FC,&h0050,&h0001,&hC020,&h0100,&h0403,&hC010,&hFFFF:REM 2
1620 DATA &h11FC,&h0050,&h0001,&hC020,&h0100,&h0403,&hC010,&hFE7F:REM 3
1630 DATA &h0080,&h2006,&h8120,&h4208,&h2208,&hFF10,&h0407,&hFE20:REM 4
1640 DATA &h0FFF,&h0040,&hF002,&h001F,&h1001,&h0100,&h1014,&hFE7F:REM 5
1650 DATA &h41F0,&h0100,&h1004,&hFF80,&h2004,&h0100,&h1014,&hFE7F:REM 6
1660 DATA &h07FF,&h0020,&h0001,&h0040,&h0401,&h4010,&h0100,&hFE04:REM 7
1670 DATA &h11FC,&h0050,&h1803,&hFF40,&h3005,&h0100,&h1014,&hFE7F:REM 8
1680 DATA &h11FC,&h0050,&h2003,&hFE40,&h1003,&h0040,&h0401,&hFE1F:REM 9
1690 DATA &h11C3,&h0491,&h2025,&h0941,&h504A,&h1282,&hD114,&hFE71:REM 10
1700 DATA &h0700,&h0010,&h0400,&h0020,&h2001,&h0140,&h0012,&hFE3F:REM Jack
1710 DATA &h11FC,&h0050,&h1803,&h00C0,&h3106,&h0190,&h1015,&hFF7F:REM Queen
1720 DATA &h0E01,&h184C,&h7032,&h1001,&h2104,&h0110,&h5009,&hFF00:REM King

```


ค โปรแกรม . Racing

```

10 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
20 REM XXX XXX
30 REM XXX RACING XXX
40 REM XXX Copyright 1979 (C) XXX
50 REM XXX by XXX
60 REM XXX ASCII Laboratories XXX
70 REM XXX XXX
80 REM XXX NAC XXX
90 REM XXX XXX
100 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
110 GOTO 670
120 FOR I=0 TO 4:D(I)=-1:NEXT
130 FOR I=4 TO 0 STEP -1
140 IF RND(1)<ST(1) THEN 500
150 X=X(I):Y=Y(I):VX=0:VY=0
160 ON
170 IF FNP(X+1,Y+1)=32 THEN VX=1:VY=1
180 IF FNP(X+1,Y)=32 THEN VX=1:VY=0
190 IF FNP(X+1,Y-1)=32 THEN VX=1:VY=-1
200 IF X>30 THEN C(I)=2
210 GOTO 470
220 IF FNP(X+1,Y-1)=32 THEN VX=1:VY=-1
230 IF FNP(X,Y-1)=32 THEN VX=0:VY=-1
240 IF FNP(X-1,Y-1)=32 THEN VX=-1:VY=-1
250 IF Y<8 THEN C(I)=3
260 GOTO 470
270 IF FNP(X-1,Y-1)=32 THEN VX=-1:VY=-1
280 IF FNP(X-1,Y)=32 THEN VX=-1:VY=0
290 IF FNP(X-1,Y+1)=32 THEN VX=-1:VY=1
300 IF X<7 THEN C(I)=4
310 GOTO 470
320 IF FNP(X-1,Y+1)=32 THEN VX=-1:VY=1
330 IF FNP(X,Y+1)=32 THEN VX=0:VY=1
340 IF FNP(X+1,Y+1)=32 THEN VX=1:VY=1
350 IF Y>17 THEN C(I)=5
360 GOTO 470
370 IF FNP(X+1,Y+1)=32 THEN VX=1:VY=1
380 IF FNP(X+1,Y-1)=32 THEN VX=1:VY=-1
390 IF FNP(X+1,Y)=32 THEN VX=1:VY=0
400 IF X>20 THEN C(I)=6:BEEP:GOTO 420
410 GOTO 470
420 LOCATE X,Y:PRINT " ";
430 FOR J=0 TO 4:IF D(J)<0 THEN 440 ELSE NEXT J
440 LOCATE 19,12+J:COLOR7:PRINT USING"# place ";J+1;:COLOR I+2:PRINT"0";
450 PRINT USING"#";5-I;:D(J)=5-I:IF J=4 THEN 520
460 GOTO 500
470 LOCATE X,Y:COLOR7:PRINT " ";:
480 X(I)=X+VX:Y(I)=Y+VY
490 LOCATE X(I),Y(I):COLORI+2:PRINT"0";
500 NEXT I
510 GOTO 130
520 IF D(0)=K1 AND D(1)=K2 OR D(0)=K2 AND D(1)=K1 THEN W1=BE*KA(K1)*KA(K2):GOTO
540
530 IF D(0)=K1 AND K2=0 THEN W1=BE*KA(K1):GOTO 560
540 LOCATE 10,11:PRINT "XXX You lose XXX";
550 GOTO 580
560 LOCATE 10,11:PRINT "XXX You win XXX";
570 MA=MA+W1
580 IF MA=0 THEN LOCATE 10,2:PRINT"XXX GAME OVER XXX";:GOTO 660
590 FOR I=0 TO 4
600 KA(D(I))=KA(D(I))+(1-2)/10
610 NEXT
620 LOCATE 10,13:PRINT"You have";
630 LOCATE 10,14:PRINT USING"#####";MA;
640 A$=INPUT$(1)
650 GOTO 720
660 LOCATE 12,5:PRINT"Replay (y/n)?"
662 A$=INKEY$:IF A$="y" THEN RUN
663 IF A$="Y" THEN RUN
664 IF A$="n" THEN PRINT CHR$(12):WIDTH 80,25:FILES:END
665 IF A$="N" THEN PRINT CHR$(12):WIDTH 80,25:FILES:END
666 GOTO 660
670 WIDTH 40,25:CONSOLE0,25,0,1
680 DEFFNPK(X,Y)=PEEK(&HF302+X+X*Y*120)
690 COLOR7,32,0
700 FOR I=1 TO 5:KA(I)=4.75-.25*I:NEXT:KA(0)=1
710 MA=1000
720 REM *** bet ***
730 COLOR7
740 PRINT CHR$(12);"XXX RACING XXX"

```



```

740 PRINT CHR$(12);"XXX RACING XXX"
750 PRINT:PRINT:PRINT "BI list (pay list of 2 numbers which "
760 PRINT "      enter 1st & 2nd)"
770 PRINT
780 PRINT"\ | 1 | 2 | 3 | 4 | 5 |"
790 PRINT"-----"
800 PRINT"-|";FOR I1=1 TO 5:PRINT USING"##.#|";KA(I1);NEXT:PRINT
810 FOR I=1 TO 4
820 PRINT"-----"
830 PRINT USING"##.#|";I;
840 FOR J=1 TO 5
850 IF I<J THEN PRINT USING"##.#|";KA(I)*KA(J); ELSE PRINT" - |";
860 NEXT J
870 PRINT
880 NEXT I
890 PRINT"-----"
900 PRINT:PRINT USING "You have#####";MA
910 INPUT"Bet how much";BET
920 IF BET>MA THEN PRINT "you have";MA;"baht only!!!!":GOTO 910
930 MA=MA-BET
940 PRINT"Want to select BI list?(y or n)";
950 A$=INPUT$(1):IF A$="y" THEN 1020
960 IF A$<>"n" THEN 950
970 PRINT "N"
980 INPUT"what No.";K1
990 IF K1<1 OR K1>5 THEN 980
1000 K2=0
1010 GOTO 1060
1020 PRINT"Y"
1030 INPUT"Enter BI list (A,B)";K1,K2
1040 IF K1=K2 OR K1<1 OR K1>5 THEN 1030
1050 IF K2<1 OR K2>5 THEN 1030
1060 COLOR 7
1070 PRINT CHR$(12);"XX RACING XX";
1080 COLOR6:PRINT" 10 ";
1090 COLOR5:PRINT" 20 ";
1100 COLOR4:PRINT" 30 ";
1110 COLOR3:PRINT" 40 ";
1120 COLOR2:PRINT" 50 ";
1130 COLOR 7
1140 LINE(1,1)-(35,1),"-"
1150 LOCATE 36,1:PRINT" ";
1160 LINE(36,2)-(36,23),"|"
1170 LOCATE 36,24:PRINT"J";
1180 LINE(35,24)-(1,24),"-"
1190 LOCATE 0,24:PRINT" ";
1200 LINE(0,23)-(0,2),"|"
1210 LINE(24,19)-(24,23),"|"
1220 LOCATE 0,1:PRINT" ";
1230 LINE(9,7)-(27,7),"█"
1240 LINE(9,18)-(27,18),"█"
1250 LINE(6,10)-(6,15),"█"
1260 LINE(30,10)-(30,15),"█"
1270 LOCATE 21,18:PRINT" |";
1280 LINE(6,9)-(8,7),"▲"
1290 LINE(7,9)-(8,8),"▲"
1300 LINE(6,16)-(8,18),"▲"
1310 LINE(7,16)-(8,17),"▲"
1320 LINE(30,9)-(28,7),"▲"
1330 LINE(29,9)-(28,8),"▲"
1340 LINE(30,16)-(28,18),"▲"
1350 LINE(29,16)-(28,17),"▲"
1360 LOCATE 9,9:PRINT"BET";BET;"No.";K1;
1370 IF K2>0 THEN PRINT" ";K2;
1380 LOCATE 10,10:PRINT"rate";KA(K1)*KA(K2);
1390 RESTORE
1400 FOR I=0 TO 4:READ X(I),Y(I),C(I),ST(I)
1410 NEXT
1420 GOSUB 1450
1430 GOTO 120
1440 DATA 25,23,1,.1,25,22,1,.11,25,21,1,.12,25,20,1,.13,25,19,1,.14
1450 REM xxx write colors xxx
1460 FOR I=0 TO 4
1470 LOCATE X(I),Y(I):COLOR I+2
1480 PRINT"0";
1490 NEXT:RETURN
1500 DATA 13,-3,57,44,1,9,-7,-9,-2,1,12,-84,17,8,-2,2,-25,8,27,-26,-28,-3,22
1510 DATA 66,23,-89,33,18,-16,6,0,-41,44,21,1,13,3,-17,19,-5,3,-9,-4,14,-102,-3,
22,0,0,51,-37
1520 DATA 32,19,10,-10,9,-1,4,-12,-84,-3

```


ง โปรแกรม Demo

```

100 REM *****
110 REM *** XXX
120 REM *** NEC PC-8001 XXX
130 REM *** BASIC GAME BOOK XXX
140 REM *** XXX
150 REM *** Copyright 1979 (C) XXX
160 REM *** by XXX
170 REM *** ASCII Laboratories XXX
180 REM *** XXX
190 REM *** NAC & JUN XXX
200 REM *** XXX
210 REM *****
220 CLEAR 3000 :DEFINT A-Z
230 DIM A$(4,5),AS(100),PC(200),BA(100),GA(100),BO(100),AM(100)
240 FOR I=0 TO 4
250 FOR J=0 TO 5
260 READ D$
270 A$(I,J)=D$+STRING$(LEN(D$),CHR$(29))+CHR$(31)
280 NEXT:NEXT
290 CONSOLE0,25,0,1:WIDTH 80,25:COLOR7,32,0:PRINT CHR$(12);
300 CO=0:X=0:Y=0
310 I=0:GOSUB 700:GET$(0,0)-(29,5),AS:PRINT CHR$(12)
320 I=1:GOSUB 700:GET$(0,0)-(43,5),PC:PRINT CHR$(12)
330 I=2:GOSUB 700:GET$(0,0)-(32,5),BA:PRINT CHR$(12)
340 I=3:GOSUB 700:GET$(0,0)-(29,5),GA:PRINT CHR$(12)
350 I=4:GOSUB 700:GET$(0,0)-(29,5),BO:PRINT CHR$(12)
360 READ RE
370 IF RE=0 THEN 420
380 IF RE=1 THEN 470
390 IF RE=8 THEN COLOR7:PRINT CHR$(12);:GOSUB 900:GOTO 360
400 IF RE=9 THEN FOR I=0 TO 30000!:NEXT:RUN
410 GOTO 360
420 READ I:READ CO:READ WA
430 READ X:IF X<0 THEN 360 ELSE READ Y:IF Y<0 THEN 450
440 GOSUB 800:FOR L=0 TO WA:NEXT:L:GOSUB 790 :GOTO 430
450 Y=ABS(Y):IF Y>25 THEN Y=0
460 GOSUB 800:GOTO 360
470 FOR K=0 TO 3:FOR I=0 TO 24:GOSUB 900
480 LINE I,2:FOR J=0 TO 50:NEXT:LINE I,0
490 NEXT :NEXT
500 FOR II=0 TO 2: FOR I=0 TO 50
510 FOR J=I TO I-20 STEP -2:IF J>=0 AND J<25 THEN LINE J,2
520 GOSUB 900:NEXT J
530 FOR J=I TO I-20 STEP -2:IF J>=0 AND J<25 THEN LINE J,0
540 NEXT
550 NEXT I:NEXT II
560 FOR KT=0 TO 2
570 FOR I=0 TO 24:LINE I,2:FOR K=0 TO 10:GOSUB 900:NEXT:NEXT
580 GOSUB 880
590 FOR I=0 TO 24:LINE I,0:FOR K=0 TO 10:GOSUB 900:NEXT:NEXT
600 GOSUB 880
610 FOR I=0 TO 5:FOR J=0 TO 24:LINE J,2:NEXT:GOSUB 900
620 FOR J=0 TO 24:LINE J,0:NEXT:GOSUB 900:NEXT
630 NEXT KT
640 FOR I=0 TO 100:GOSUB 900:NEXT
650 FOR I=0 TO 10 STEP 6:GOSUB 900
660 FOR J=I TO I+5:LINE J,1:NEXT:BEEP:GOSUB 890
670 FOR J=I TO I+5:LINE J,0:NEXT:GOSUB 880
680 NEXT I
690 GOTO 360
700 COLORCO
710 LOCATE X,Y
720 FOR J=0 TO 5:PRINT A$(I,J);:NEXT
730 RETURN
740 COLOR7
750 LOCATE X,Y
760 FOR J=0 TO 5:K=(LEN(A$(I,J))-1)/2
770 PRINT SPACE$(K)+MID$(A$(I,J),K+1);
780 NEXT:RETURN
790 COLOR0:GOTO 820
800 COLOR CO
810 GOSUB 900
820 IF I=0 THEN PUT$(X,Y)-(X+29,Y+4),AS
830 IF I=1 THEN PUT$(X,Y)-(X+43,Y+5),PC
840 IF I=2 THEN PUT$(X,Y)-(X+32,Y+4),BA
850 IF I=3 THEN PUT$(X,Y)-(X+29,Y+4),GA
860 IF I=4 THEN PUT$(X,Y)-(X+29,Y+4),BO
870 RETURN
880 FOR KK=0 TO 100:GOSUB 900:NEXT:RETURN
890 FOR KK=0 TO 500:GOSUB 900:NEXT:RETURN
900 IF INKEY$="" THEN RETURN

```



```

900 IF INKEY="" THEN RETURN
910 WIDTH80,25:COLOR 7,0,0:PRINT CHR$(12)
913 LINE(0,0)-(79,4),"=",B
915 LINE(0,4)-(79,20),"=",B
916 LINE(0,20)-(79,24),"=",B
920 LOCATE 25,2:COLOR 5:PRINT "BASIC GAME BOOK DISK No.1"
940 LOCATE 10,6:COLOR 3:PRINT "a: DEMO"
950 LOCATE 10,7:PRINT "b: FORMAT"
960 LOCATE 10,8:PRINT "c: BACKUP"
970 LOCATE 10,9:PRINT "d: RACE"
980 LOCATE 10,10:PRINT "e: B-JACK"
990 LOCATE 10,11:PRINT "f: BREAK"
1000 LOCATE 10,12:PRINT "g: D-MAZE"
1010 LOCATE 10,13:PRINT "h: GOLF"
1020 LOCATE 10,14:PRINT "i: BOWL"
1030 LOCATE 10,15:PRINT "j: PLOT"
1040 LOCATE 10,16:PRINT "k: GRAPH2"
1041 LOCATE 10,17:PRINT "l: GRAPH3"
1042 LOCATE 10,18:PRINT "m: EXIT"
1045 LOCATE 15,22:COLOR 6:PRINT "SELECT GAME (PLEASE HIT CHARACTER)" ;
1050 A%=INKEY$:IF A%="" THEN 1050
1051 V%=ASB(A%):X%=V%-96:ON
1052 RUN"demo"
1053 RUN"format"
1054 RUN"backup"
1055 RUN"race"
1056 RUN"b-jack"
1057 RUN"break"
1058 RUN"d-maze"
1059 RUN"golf"
1060 RUN"bowl"
1061 RUN"plot"
1062 RUN"graph2"
1063 RUN"graph3"
1064 PRINT CHR$(12):CONSOLE 0,25,1,1:WIDTH 80,25:END
1070 DATA
1080 DATA "ASCII"
1090 DATA "ASCII"
1100 DATA "ASCII"
1110 DATA "ASCII"
1120 DATA "ASCII"
1130 DATA "ASCII"
1140 DATA "PC-8001"
1150 DATA "PC-8001"
1160 DATA "PC-8001"
1170 DATA "PC-8001"
1180 DATA "PC-8001"
1190 DATA "PC-8001"
1200 DATA "BASIC"
1210 DATA "BASIC"
1220 DATA "BASIC"
1230 DATA "BASIC"
1240 DATA "BASIC"
1250 DATA "BASIC"
1260 DATA "GAME"
1270 DATA "GAME"
1280 DATA "GAME"
1290 DATA "GAME"
1300 DATA "GAME"
1310 DATA "BOOK"
1320 DATA "BOOK"
1330 DATA "BOOK"
1340 DATA "BOOK"
1350 DATA "BOOK"
1360 DATA "BOOK"
1370 DATA 0,0,0,2,0,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10
1380 DATA 11,9,12,0,13,7,14,6,15,5,16,4,17,3,18,2,19,1,20,0
1390 DATA 19,0,18,0,17,0,16,0,15,0,14,0,13,0,12,0,11,0,10,0,9,0,8,0,7,0,6,0,5,0,
4,-30
1400 DATA 0,1,7,1,31,18,28,17,26,16,24,15,23,14,22,13,22,12
1410 DATA 21,11,20,10,20,9,20,8,20,7,20,-6
1420 DATA 0,2,4,1,0,18,1,18,2,17,3,16,4,15,5,14,6,13,7,-12
1430 DATA 0,3,6,1,40,18,39,18,37,18,36,18,35,18,34,18,33,18,32,18,31,18
1440 DATA 30,18,29,18,27,18,26,18,25,18,24,18,23,18,22,18,21,18,20,18,19,18
1450 DATA 17,18,16,18,15,18,14,18,13,18,12,18,11,-18
1460 DATA 0,4,6,1,49,12,48,12,47,12,46,12,45,12,44,12,43,12
1470 DATA 43,13,43,14,43,15,43,16,43,17,43,-18
1480 DATA 0,3,6,1,11,18,10,18,11,-18
1490 DATA 0,4,6,1,43,18,44,18,45,-18
1500 DATA 0,0,0,2,0,4,-30,0,1,7,1,20,-6,0,2,4,1,7,-12
1510 DATA 0,3,6,1,11,-18,0,4,6,1,45,-18,1,1,1,9
1520 DATA 13,-3,57,44,1,9,-7,-9,-2,1,12,-84,17,8,-2,2,-25,8,27,-26,-28,-3,22
1530 DATA 66,23,-89,33,18,-16,6,0,-41,44,21,1,13,3,-17,19,-5,3,-9,-4,14,-102
1540 DATA -3,22,0,0,51,-37,32,19,10,-10,9,-1,4,-12,-84,-3

```


ประวัติผู้เขียน

นายสุนทร ศรีรัฐ เกิดวันที่ 30 กันยายน 2500 ที่จังหวัดพัทลุง
สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต (ไฟฟ้า) จากคณะวิศวกรรมศาสตร์
มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2522 เข้าศึกษาระดับปริญญาโทบริหาร
ศาสตรบัณฑิต สาขาวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ ในปี พ.ศ. 2524
ปัจจุบันเป็นผู้วิเคราะห์ระบบของบริษัทไทยลย์ประกันภัย



ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย