



## เอกสารอ้างอิง

1. H. Genchi, K. Mori, S. Watanabe, S. Katsuragi, "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting" Proc. of the IEEE, Vol. 56, No. 8, PP. 1292 - 1301, Aug. 1968
2. J.R. Ullmann, "Advance in Character Recognition" Application of Pattern Recognition, ed. by K.S. Fu, New York: Crc Press, 1982
3. T. Pavlidis, F. Ali, "Computer Recognition of Handwritten Numerals by polygonal Approximations" IEEE Trans. Syst., Man, Cybern., Vol. SMC - 5, PP. 610 - 614, Nov. 1975
4. T. Pavlidis, F. Ali, "Syntactic Recognition of Handwritten Numerals" IEEE Trans. Syst., Man, Cybern., Vol. SMC - 7, PP. 537 - 541, July 1977
5. P. Hiranvanichakorn, T. Agui, M. Nakajima, "A Recognition Method of Thai Characters" Trans. IEEE Japan, Vol. E65, No. 12, PP. 737 - 744, Dec. 1982
6. S.Y. Lu, K.S. Fu, "Error - Correcting Tree Automata for Syntactic Pattern Recognition" IEEE Trans. Comput., Vol. C - 27, PP. 1040 - 1053, Nov. 1978
7. Suen, C.Y., Berthod, M. and Mori, S. "Automatic Recognition of Handprinted Characters - The state of the art" Proc. of the IEEE, Vol. 68, No. 4, PP. 469 - 487, 1980
8. L.G. Shapiro, "A structural Model of Shape" IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI - 2, PP. 111 - 126, March 1980

## เอกสารอ้างอิง (ต่อ)

9. ชม กัมปาน "การจดจำรูปแบบตัวพิมพ์อักษรไทยโดยวิธีการเปรียบเทียบ" เอกสารรวมบทความการประชุมวิชาการวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา ครั้งที่ 5 ณ มหาวิทยาลัยเชียงใหม่ 1 - 2 พฤศจิกายน 2525
10. สุรพันธ์, กิตติพล, มนัส, ชม "เครื่องอ่านรูปแบบสำหรับการจำแนกรูปแบบตัวอักษรด้วยไมโครคอมพิวเตอร์" เอกสารรวมบทความการประชุมวิชาการวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา ครั้งที่ 8 คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าวิทยาเขตพระนครเหนือ 19 - 20 ธันวาคม 2528
11. พิเชฐ, กฤษดา "การสร้างระบบอ่านตัวอักษรแบบง่าย ๆ โดยใช้กล้องวิดีโอคอน" เอกสารรวมบทความการประชุมวิชาการวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา ครั้งที่ 8 คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าวิทยาเขตพระนครเหนือ 19 - 20 ธันวาคม 2528
12. K.S. Fu, Digital Pattern Recognition, 2nd ed., Berlin: Springer - Verlag, 1980
13. J.R. Ullmann, Pattern Recognition Techniques, London: Butterworth & Co., 1973
14. T. Pavlidis, Structural Pattern Recognition, Berlin: Springer - Verlag, 1977
15. L.P. Horwitz, G.L. Shelton, "Pattern Recognition Using Autocorrelation" Proc. of the IRE, Jan. 1961
16. T. Tricket, "Designing Standard OCR Fonts" Optical Character Recognition and the Years Ahead, The Business Press, Elmhurst, Ill., PP. 370 - 385. 1969
17. K.S. Fu, Syntactic Pattern Recognition, Applications, Berlin: Springer - Verlag, 1977



## เอกสารอ้างอิง (ต่อ)

18. M.J. Nilsson, Learning Machines, New York: Mc Graw - Hill, 1965
19. K. Fukunaga, Introduction to Statistical Pattern Recognition,  
New York: Academic Press, 1972
20. H.C. Andrews, Introduction to Mathematical Techniques in Pattern Recognition, New York : Wiley, 1972
21. C.H. Chen, Statistical Pattern Recognition, Washington, D.C. :  
Hayden Book Co., 1973
22. K.S. Fu, Syntactic Methods in Pattern Recognition, New York :  
Academic Press, 1974
23. J.J. Leimer, "Design Factors in the Development of an Optical  
Character Recognition Machine" IRE Trans. Inform. Theory,  
Vol. IT - 8, PP. 167 - 171, 1962
24. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations"  
IRE Trans. Elec. Com., Vol. EC - 10, PP. 260 - 268, 1961
25. T. Paylidis, "Segmentation of Plane Curves" IEEE Trans. Comput.,  
Vol. C - 23, No. 8, PP. 860 - 870, Aug. 1974
26. P.J. Gray, "Optical Scanning, OCR, and MICR" Automatic Data  
Processing Hand Book, ed. By the Diebold Group Inc., New  
York : Mc Graw - Hill, 1977
27. A.J. Atrubin, "Character - recognition Systems" Encyclopedia of  
Instrumentation and Control, ed. by D.M. Considine, New  
York : Mc Graw - Hill, 1971
28. BRITISH COMPUTER SOCIETY, Character Recognition 1967
29. M.R. Bartry, "The IBM 1975 Optical Page Reader Part II : Video  
Thresholding System" IBM J. Res. Develop., Vol. 12,  
No. 5, PP. 354 - 363, 1968

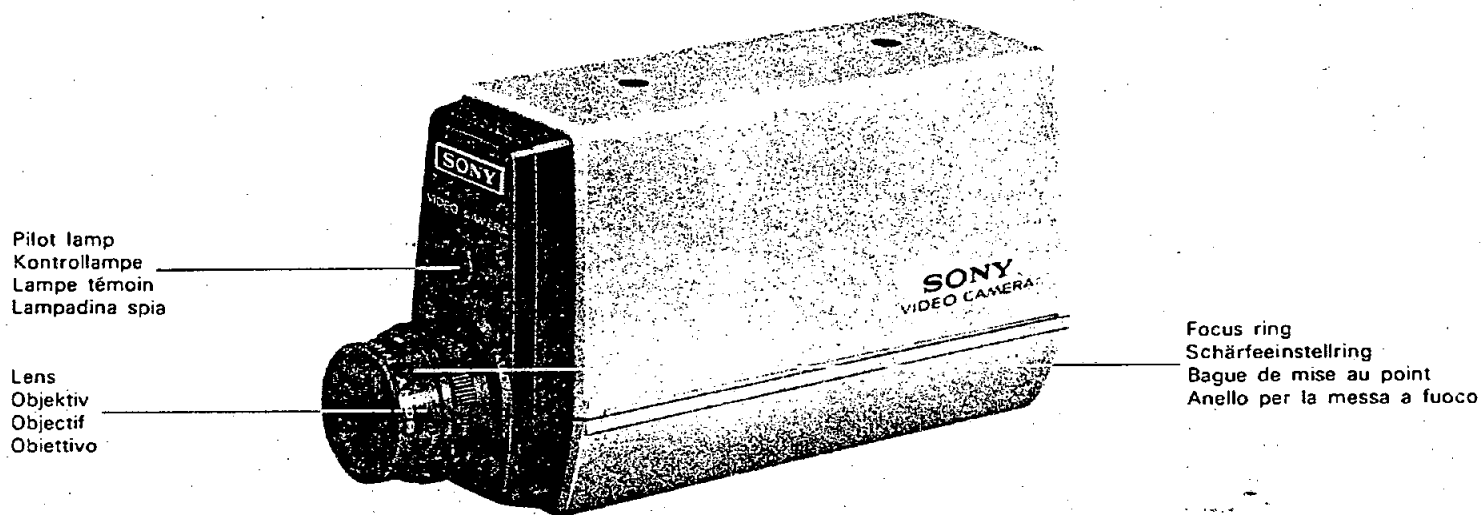
## เอกสารอ้างอิง (ต่อ)

30. M. Keryan, "Build a Video Digitizer" Byte, Nov, 1982 : 174 - 188
31. F. Lawson, "Photography of insects" Photographic Techniques in Scientific Research Vol. I, ed. by John Guise and A.A. Newman, London : Academic Press, 1973
32. Stroebel, View Camera Technique, London : Focal Press, 1976
33. Brice and Fennema, "Scene Analysis" Artificial Intelligence, Vol. 1, PP. 205 - 226, 1970
34. National Semiconductor, Linear Data Book, 1981
35. Grob, Basic Television principles and servicing, 4th ed., Tokyo : Mc Graw - Hill, 1985
36. Microsoft Inc., TASC Applesoft Basic Compiler
37. Apple Computer, Apple II Reference Manual.

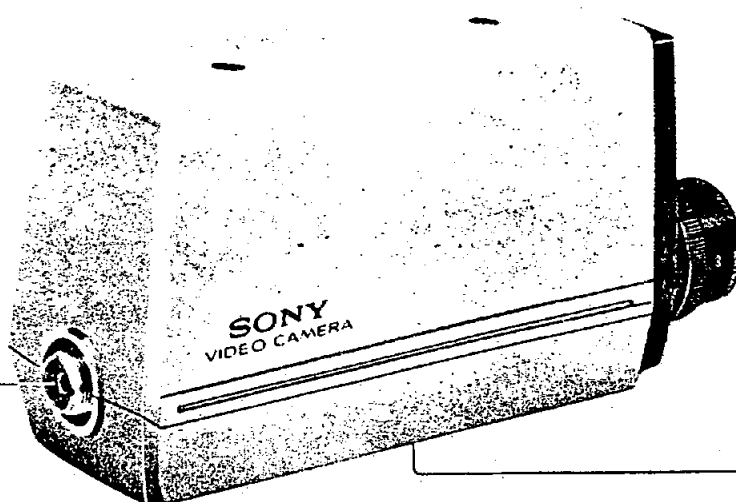
**ภาคผนวก**

ภาคผนวก ก. รายละเอียดและคุณสมบัติของกล่องวิติคอนที่ใช้

LOCATION OF PARTS / GERÄTEANSICHT / EMBLACEMENT DES SOUS-ENSEMBLES /  
POSIZIONE DELLE PARTI



Camera cable connector  
 Anschluß für Verbindungskabel  
 Connecteur coaxial de câble  
 de caméra  
 Connettore per il cavo della  
 telecamera



Tripod receptacle  
 Stativgewinde  
 Taraudage de fixation de pie  
 Ricettacolo per il treppiede

## SPECIFICATIONS

Pick-up tube	2/3-inch vidicon, electrostatic focus, electromagnetic deflection
Scanning system	625 lines/frame, 25 frames/sec.
Sync system	internal 2:1 interlace
Horizontal resolution	more than 400 lines at center
Horizontal frequency	15.625 kHz
Vertical frequency	50 Hz
Minimum illumination	10 footcandles (100 lux)
Automatic sensitivity control range	10-10,000 footcandles (100-100,000 lux)
Video output	1.0 Vp-p, sync negative, 75 ohms, unbalanced, UHF-type connector
Signal-to-noise ratio	better than 42 dB
Power requirements	17 V DC (power supplied by the PU-1400CE or PU-1401CE)
Power consumption	3.5 W
Operating temperature	-5°C to 45°C (23°F to 113°F)
Lens mount	Standard C-mount
Dimensions	approx. 2 1/2" x 4" x 8" (64 x 101 x 203 mm) (w/h/d) including lens
Weight	approx. 2 lb 2 oz (0.98 kg) including lens

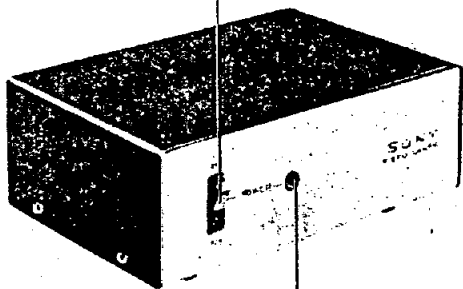


LOCATION OF PARTS/LAGE DER TEILE/EMPLACEMENT DES SOUS-ENSEMBLES/POSIZIONE DELLE PARTI

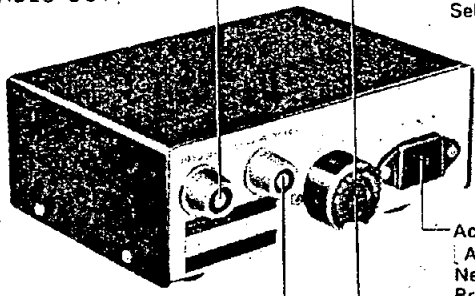
Power switch [POWER]  
 Netzschalter [POWER]  
 Interrupteur d'alimentation [POWER]  
 Interruttore di alimentazione [POWER]

Video output connector [VIDEO OUT]  
 Videoausgang [VIDEO OUT]  
 Connecteur coaxial de sortie vidéo [VIDEO OUT]  
 Connettore di uscita video [VIDEO OUT]

Voltage selector  
 Spannungswähler  
 Sélecteur de tension  
 d'alimentation  
 Selettore di tensione



Pilot lamp  
 Kontrollampe  
 Lampe témoin  
 Lampadina spia

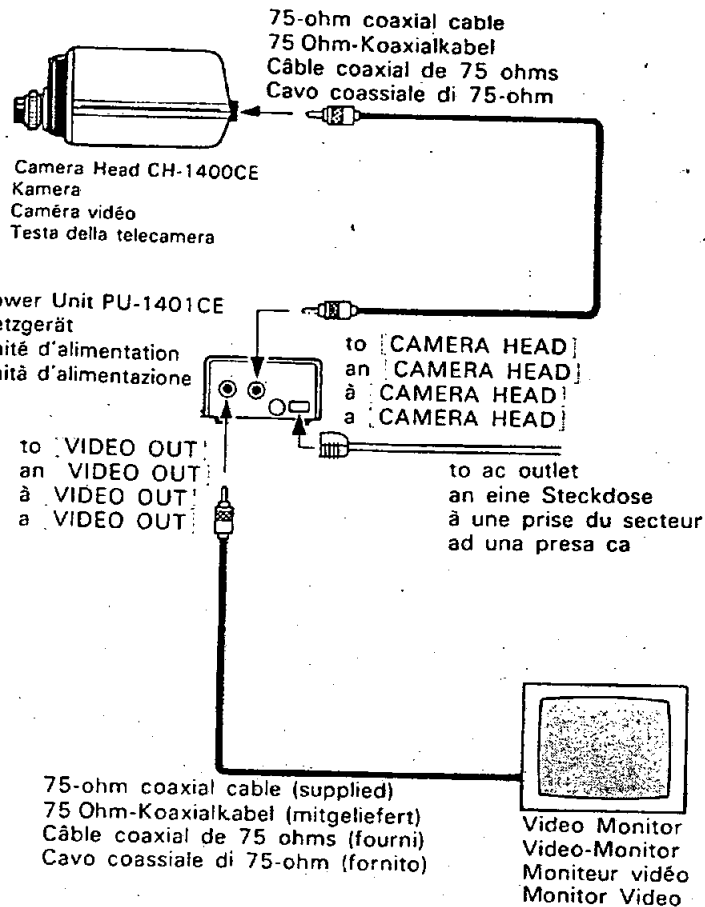


Ac input connector  
 [AC IN]  
 Netzanschluß [AC IN]  
 Prise d'alimentation  
 CA [AC IN]  
 Ingresso ca [AC IN]

Fuse holder  
 Sicherungshalter  
 Porte-fusible  
 Portafusibili

Camera head connector [CAMERA HEAD]  
 Kameraanschluß [CAMERA HEAD]  
 Connecteur coaxial de caméra  
 [CAMERA HEAD]  
 Connettore per la testa della telecamera  
 [CAMERA HEAD]

## CONNECTIONS/ANSCHLUSS/CONNEXIONS/COLLEGAMENTI



Note: Cable between the Camera Head and the Power Unit can be extended up to:

- 164 feet (50 m) with 3C2V coaxial cable
- 460 feet (140 m) with 5C2V coaxial cable
- 787 feet (240 m) with 7C2V coaxial cable

Hinweis: Das Kabel zwischen Kamera und Netzgerät kann wie folgt verlängert werden.

- 50 m mit Hilfe des 3C2V-Koaxialkabels
- 140 m mit Hilfe des 5C2V-Koaxialkabels
- 240 m mit Hilfe des 7C2V-Koaxialkabels

Note: La longueur maximum du câble entre la caméra et l'unité d'alimentation est de:

- 50 m avec le câble coaxial 3C2V
- 140 m avec le câble coaxial 5C2V
- 240 m avec le câble coaxial 7C2V

Nota: Il cavo fra la testa della telecamera e l'unità d'alimentazione può essere allungato:

- fino a 50 m con cavo coassiale 3C2V,
- fino a 140 m con cavo coassiale 5C2V,
- fino a 240 m con cavo coassiale 7C2V.

## OPERATION

1. Complete the connections as illustrated.
2. Set the POWER switch to ON. The pilot lamps will light indicating that the camera head and power unit are operating.
3. Allow the camera head to warm up for 30 seconds.
4. Remove the lens cap.
5. Adjust the lens focus ring of the camera head to produce a clear, sharp picture on the monitor screen.

## SPECIFICATIONS

Video output	1.0 Vp-p, sync negative, 75 ohms, unbalanced, UHF-type connector
Power requirements	110, 127, 220, 240V ac~ adjustable, 50 or 60 Hz (240V ac~, 50 Hz for United Kingdom)
Power consumption	14 W
Operating temperature	23°F - 113°F (-5°C to +45°C)
Dimensions	Approx. 6 3/4"(w) x 2 3/8"(h) x 5 1/16"(d) (171 x 60 x 129 mm) including projecting parts and controls
Weight	Approx. 3 lb 1 oz (1.45 kg)
Accessory supplied	5 ft (1.5 m) 75-ohm coaxial cable with UHF-type connectors

While the information given is true at the time of printing, small production changes in following our company's policy of improvement through research and design might not necessarily be indicated in the specifications. We would ask you to check with your appointed Sony dealer if clarification on any point is required.

## ภาคผนวก ข. PROGRAM LISTING

:ASM

```

0100 *-----
0110 *PROGRAM FOR READING DATA FROM
0120 *CAMERA
0130 *-----
0140      .OR $300
0150 *-----
0160 * X-STORE SLOT # ; Y-STORE COUNTER
0170 * #FC-STORE LINE COUNTER
0180 * #FD-STORE LOOP LINE COUNTER
0190 * #FE-LOW BYTE ADDRESS BUFFER
0200 * #FF-HIGH BYTE ADDRESS BUFFER
0210 * #FB-STORE BLOCK COUNTER
0220 *-----
0300: A9 0A 0230 MAIN LDA #10 ;SET STARTING ADDRESS
0302: B5 F9 0232 STA #F9
0304: A9 00 0238 LDA #00 ;SET STARTING ADDRESS
0306: B5 FE 0240 STA #FE
0308: A9 20 0250 LDA #20
030A: B5 FF 0260 STA #FF
030C: A0 00 0270 LDY #00 ;CLEAR COUNTER
030E: B4 FC 0280 STY #FC ;CLEAR LINCNT
0310: B4 FB 0290 STY #FB ;CLEAR BLOCK COUNTER
0312: A5 ED 0300 LDA #ED ;LOAD BLCK NUMBER(n)
0314: 0A 0310 ASL
0316: 0A 0320 ASL
0318: 0A 0330 ASL
0317: 0A 0340 ASL ;X=(n0)
0318: AA 0350 TAX
0319: BD 80 CO 0360 LDA #C080,X ;SEND BKCLR
031C: BD B3 CO 0370 LDA #C0B3,X ;CLEAR BUFFER
031F: BD 81 CO 0380 CCLE LDA #C0B1,X
0322: C6 F9 0382 DEC #F9
0324: D0 F9 0390 BNE CCLE
0326: A9 C0 0400 START LDA #192 ;SET LINE COUNTER
0328: B5 FD 0410 STA #FD
032A: A9 14 0420 LDA #20 ;IGNORE STARTING LINE
032C: B5 FA 0430 STA #FA
032E: A9 80 0440 CHKVSY LDA #80 ;CHECK VSY=1?
0330: 3D B2 CO 0450 AND #C0B2,X
0333: F0 F9 0460 BEQ CHKVSY
0335: A9 01 0470 RDY LDA #01 ;CHECK DATA READY
0337: 3D B2 CO 0480 AND #C0B2,X
033A: F0 F9 0490 BEQ RDY
033C: A9 80 0500 EXTEND LDA #80 ;CHECK VSY=1?
033E: 3D B2 CO 0510 AND #C0B2,X
0341: F0 F9 0520 BEQ EXTEND
0343: A9 01 0530 CANC LDA #01 ;CHECK DATA READY
0345: 3D B2 CO 0540 AND #C0B2,X
0348: F0 F9 0550 BEQ CANC
034A: BD B3 CO 0560 LDA #C0B3,X
034D: C6 FA 0570 DEC #FA
034F: D0 F2 0580 BNE CANC
0351: A9 01 0590 DRDY LDA #01 ;CHECK DATA READY
0353: 3D B2 CO 0600 AND #C0B2,X
0356: F0 F9 0610 BEQ DRDY
0358: BD 84 CO 0620 SAVE LDA #C0B4,X ;LOAD DATA
0359: 49 00 0630 EOR #00 ;DELAY
035D: 91 FE 0640 STA (MEM),Y
035F: BD B3 CO 0650 LDA #C0B3,X ;CLEAR BUFFER
0362: C8 0660 INY ;INCREMENT LINE COUNTER
0363: C6 FD 0670 CHK192 DEC #FD
0365: D0 EA 0680 BNE DRDY
0367: 18 0690 CLC
0368: A5 FE 0700 LDA #FE ;SET NEW BLOCK MEMDRY
036A: 69 C0 0710 ADC #C0
036C: B5 FE 0720 STA #FE
036E: A5 FF 0730 LDA #FF
0370: 69 00 0740 ADC #00
0372: B5 FF 0750 STA #FF
0374: A0 00 0760 INC LDY #00 ;CLEAR LINE COUNTER
0376: E6 FB 0770 INC #FB ;INCREMENT BLCK COUNTER
0378: BD B1 CO 0780 LDA #C0B1,X ;BLOCK INCREMENT
0378: A3 FB 0790 CHK36 LDA #FB
037D: 38 0800 SEC
037E: E9 24 0810 SBC #24
0380: D0 A4 0820 BNE START
0382: 60 0830 RTS
00FE: MEM .ED #0FE
0850 .EN

```

## SYMBOL TABLE

```

0343: CANC
031F: CCLE
0363: CHK192
0378: CHK36
032E: CHKVSY
0351: DRDY
033C: EXTEND
0374: INC
0300: MAIN
00FE: MEM
0335: RDY
0358: SAVE
0326: START

```

0000 ERRORS IN ASSEMBLY

1 ASM

```

0100 *-----
0110 * TRANSFER DATA FROM DATA BUFFER
0120 * TO HGR BUFFER AND ALINE
0130 *-----
0140      .DR $6200
0150 *-----
0160 * X-LINE COUNTER ; Y-BLOCK COUNTER
0170 * $FA-TEMPOLARY REGISTER
0180 * $FC-RESTROE LINE COUNTER
0190 *-----
6200: A9 00 0200 TRF   LDA #300
6202: AB 00 0210     TAY
6203: B5 FE 0220     STA #FE      ;SET LOW ADDRESS
6205: B5 FB 0230     STA #FB      ;CLEAR TEMP
6207: A9 20 0240     LDA #20
6209: B5 FF 0250     STA #FF
620B: A9 00 0280 BEB   LDA #300
620D: B5 FC 0290     STA #FC      ;CLEAR LINE COUNTER
620F: A9 07 0292 LODP  LDA #7
6211: B5 FD 0294     STA #FD
6213: A2 00 0300     LDX #300
6215: A1 FE 0310     LDA (MEM,X) ;LOAD DATA
6217: B5 FA 0320     STA #FA      ;STORE DATA
6219: 0A 00 0322 IBN   ASL
621A: 26 FB 0324     ROL #FB      ;SHIFT COLOR BIT
621C: 4A 00 0326     LSR
621D: 4A 00 0330 CYCLE LSR      ;CANCLE 0
621E: 26 FB 0340     ROL #FB      ;SHIFT RIGHT
6220: C6 FD 0350     DEC #FD
6222: D0 F9 0360     BNE CYCLE
6224: A5 FB 0370     LDA #FB
6226: B1 FE 0380     STA (MEM,X) ;SAVE ABIAN
6228: 18 00 0390     CLC
6229: A5 FE 0400     LDA #FE
622B: 69 01 0410     ADC #01
622D: B5 FE 0420     STA #FE
622F: 90 02 0430     BCC 6D
6231: E6 FF 0440     INC #FF
6233: A6 FC 0450 6D   LDX #FC
6235: B0 00 60 0460     LDA BUFF1,X ;LOAD LBYTE ADDRESS
6237: B5 EE 0470     STA #EE
6239: B0 00 61 0480     LDA BUFF2,X ;LOAD HBYTE ADDRESS
623B: B5 EF 0490     STA #EF
623D: A5 FA 0500     LDA #FA
623F: A5 FA 0510     STA (HGR),Y ;LOAD DATA
6241: 91 EE 0520     STA #EE      ;STORE IN HGR BUFFER
6243: EB 00 0520     INX
6245: B6 FC 0530     STX #FC      ;INCREMENT LINE COUNTER
6247: BA 00 0540 CHK192 TXA
6249: 38 00 0550     SEC
624B: E9 C0 0560     SBC #C0
624D: D0 C3 0570     BNE LODP
624F: C8 00 0580 INCL  INY
6251: 98 00 0590 CHK36 TYA
6253: 38 00 0600     SEC
6255: E9 24 0610     SBC #24
6257: D0 B9 0620     BNE BEG
6259: 60 00 0630     RTS
00FE: 0640 MEM   .EQ #00FE
00EE: 0650 HGR   .EQ #00EE
6000: 0660 BUFF1  .EQ #6000
6100: 0670 BUFF2  .EQ #6100
0680      .EN

```

SYMBOL TABLE

```

620B: BEB
6000: BUFF1
6100: BUFF2
6246: CHK192
624D: CHK36
621D: CYCLE
6233: 6D
00EE: HGR
6219: IBN
624C: INCL
620F: LODP
00FE: MEM
6200: TRF

```

0000 ERRORS IN ASSEMBLY

IASH

6000:	00	00	00	1000	.DR #6000
6003:	00	00	00	1010	.TA #800
6006:	00	00	00		
6008:	80	80	80	1020	BUFF1 .HS 0000000000000000
600B:	80	80	80		
600E:	80	80	80		
6010:	00	00	00	1030	.HS 8080808080808080
6013:	00	00	00		
6016:	00	00	00	1040	.HS 0000000000000000
6018:	80	80	80		
601D:	80	80	80		
601E:	80	80	80	1050	.HS 8080808080808080
6020:	00	00	00		
6023:	00	00	00		
6026:	00	00	00	1060	.HS 0000000000000000
6028:	80	80	80		
602B:	80	80	80		
602E:	80	80	80	1070	.HS 8080808080808080
6030:	00	00	00		
6033:	00	00	00		
6036:	00	00	00	1080	.HS 0000000000000000
6038:	80	80	80		
603B:	80	80	80		
603E:	80	80	80	1090	.HS 8080808080808080
6040:	28	28	28		
6043:	28	28	28		
6046:	28	28	28	1100	.HS 2828282828282828
6048:	AB	AB	AB		
604B:	AB	AB	AB		
604E:	AB	AB	AB	1110	.HS ABABABABABABABAB
6050:	28	28	28		
6053:	28	28	28		
6056:	28	28	28	1120	.HS 2828282828282828
6058:	AB	AB	AB		
605B:	AB	AB	AB		
605E:	AB	AB	AB	1130	.HS ABABABABABABABAB
6060:	28	28	28		
6063:	28	28	28		
6066:	28	28	28	1140	.HS 2828282828282828
6068:	AB	AB	AB		
606B:	AB	AB	AB		
606E:	AB	AB	AB	1150	.HS ABABABABABABABAB
6070:	28	28	28		
6073:	28	28	28		
6076:	28	28	28	1160	.HS 2828282828282828
6078:	AB	AB	AB		
607B:	AB	AB	AB		
607E:	AB	AB	AB	1170	.HS ABABABABABABABAB
6080:	50	50	50		
6083:	50	50	50		
6086:	50	50	50	1180	.HS 5050505050505050
6088:	00	00	00		
608B:	00	00	00		
608E:	00	00	00	1190	.HS 0000000000000000
6090:	50	50	50		
6093:	50	50	50		
6096:	50	50	50	1200	.HS 5050505050505050
6098:	00	00	00		
609B:	00	00	00		
609E:	00	00	00	1210	.HS 0000000000000000
60A0:	50	50	50		
60A3:	50	50	50		
60A6:	50	50	50	1220	.HS 5050505050505050
60A8:	00	00	00		
60AB:	00	00	00		
60AE:	00	00	00	1230	.HS 0000000000000000
60B0:	50	50	50		
60B3:	50	50	50		
60B6:	50	50	50	1240	.HS 5050505050505050
60B8:	00	00	00		
60BB:	00	00	00		
60BE:	00	00	00	1250	.HS 0000000000000000

SYMBOL TABLE

6000: BUFF1  
0000 ERRORS IN ASSEMBLY

IASH

6100:	40	44	48	0100	.DR #6100
6103:	4C	50	54		
6106:	58	5C	60	0110	BUFF2 .HS 4044484C5054585C
6108:	40	44	48		
610B:	4C	50	54		
610E:	58	5C	60	0120	.HS 4044484C5054585C
6110:	41	45	49		
6113:	4D	51	55		
6116:	59	5D	63	0130	.HS 4145494D5155595D
6118:	41	45	49		
611B:	4D	51	55		
611E:	59	5D	63	0140	.HS 4145494D5155595D
6120:	42	46	4A		
6123:	4E	52	56		
6126:	5A	5E	64	0150	.HS 42464A4E52565A5E
6128:	42	46	4A		
612B:	4E	52	56		
612E:	5A	5E	64	0160	.HS 42464A4E52565A5E
6130:	43	47	4B		
6133:	4F	53	57		
6136:	5B	5F	65	0170	.HS 43474B4F53575B5F
6138:	43	47	4B		
613B:	4F	53	57		
613E:	5B	5F	65	0180	.HS 43474B4F53575B5F
6140:	40	44	48		
6143:	4C	50	54		
6146:	58	5C	60	0190	.HS 4044484C5054585C
6148:	40	44	48		
614B:	4C	50	54		
614E:	58	5C	60	0200	.HS 4044484C5054585C
6150:	41	45	49		
6153:	4D	51	55		
6156:	59	5D	63	0210	.HS 4145494D5155595D
6158:	41	45	49		
615B:	4D	51	55		
615E:	59	5D	63	0220	.HS 4145494D5155595D
6160:	42	46	4A		
6163:	4E	52	56		
6166:	5A	5E	64	0230	.HS 42464A4E52565A5E
6168:	42	46	4A		
616B:	4E	52	56		
616E:	5A	5E	64	0240	.HS 42464A4E52565A5E
6170:	43	47	4B		
6173:	4F	53	57		
6176:	5B	5F	65	0250	.HS 43474B4F53575B5F
6178:	43	47	4B		
617B:	4F	53	57		
617E:	5B	5F	65	0260	.HS 43474B4F53575B5F
6180:	40	44	48		
6183:	4C	50	54		
6186:	58	5C	60	0270	.HS 4044484C5054585C
6188:	40	44	48		
618B:	4C	50	54		
618E:	58	5C	60	0280	.HS 4044484C5054585C
6190:	41	45	49		
6193:	4D	51	55		
6196:	59	5D	63	0290	.HS 4145494D5155595D
6198:	41	45	49		
619B:	4D	51	55		
619E:	59	5D	63	0300	.HS 4145494D5155595D
61A0:	42	46	4A		
61A3:	4E	52	56		
61A6:	5A	5E	64	0310	.HS 42464A4E52565A5E
61A8:	42	46	4A		
61AB:	4E	52	56		
61AE:	5A	5E	64	0320	.HS 42464A4E52565A5E
61B0:	43	47	4B		
61B3:	4F	53	57		
61B6:	5B	5F	65	0330	.HS 43474B4F53575B5F
61B8:	43	47	4B		
61BB:	4F	53	57		
61BE:	5B	5F	65	0340	.HS 43474B4F53575B5F

SYMBOL TABLE

6100: BUFF2  
0000 ERRORS IN ASSEMBLY



ASH

```

0100 *-----
0110 * ROUGH DECISION PROGRAM
0120 *-----
0130 .OR #4050
0140 *-----
0150 * Y-LINE COUNTER ; X-BLOCK COUNTER
0160 * #FE,#FF-STORE ADDRESS BUFFER
0170 *-----
0180 * DATA STARTING AT #2000-#3FFF
0190 * DATA IS
0200 * #FA-STORE BIT'S NUMBER
0210 * #FB-TEMPORARY BLOCK#
0220 * #FC-TEMPORARY LINE#
0230 *-----
0240 *
0250 *
0260 *
0270 * THIS IS PROGRAM TO FIND TOP
0280 *-----
4050: A9 00 0290 TOP LDA #00
4052: 85 FA 0300 STA #FA ;CLEAR BIT'S NUMBER
4054: 18 0310 CLC ;CLEAR CARRY FLAG
4055: AA 0320 TAX ;INITIALIZED CLEAR STATUS
4056: A0 05 0330 LDY #5
0340 *
4058: 98 0350 SCKTOP TYA ;CHECK END LINE
4059: 38 0360 SEC
405A: E9 25 0370 SBC #190
405C: F0 25 0380 BEQ OUT1 ;IF END GOTO OUT
405E: BD 24 40 0390 CDT1 LDA BUFF1,X ;SET STARTING ADDRESS
4061: 85 FE 0400 STA #FE
4063: BD 00 40 0410 LDA BUFF2,X
4066: 85 FF 0420 STA #FF
4068: B1 FE 0430 LDA (MEM),Y
406A: 29 7F 0440 AND #7F ;IGNORE BIT7
406C: D0 0D 0450 BNE ST1 ;IF DATA EXISTED BOTO SAVE
406E: EB 0460 INX ;INCREMENT BLOCK COUNTER
406F: 8A 0470 CEN1 TXA ;CHECK END BLOCK
4070: 38 0480 SEC
4071: E9 24 0490 SBC #36
4073: D0 E9 0500 BNE CDT1 ;IF NOT END GOTO CHECK DATA AGAIN
4075: A2 00 0510 LDX #00 ;CLEAR BLOCK
4077: C8 0520 INY
4078: 4C 38 40 0530 JMP SCKTOP
407B: 2A 0540 ST1 ROL ;CANCELLED BIT 7
407C: E6 FA 0550 LOOP1 INC #FA
407E: 2A 0560 ROL
407F: 90 FB 0570 BCC LOOP1 ;IF NOT HAVE LOGIC 0 BOTO LOOP
4081: 84 FC 0580 STY #FC
4083: 60 0590 OUT1 RTS
0600 *-----
0610 *
0620 *
0630 *
0640 * THIS IS PROGRAM TO FIND BOTTOM
0650 *-----
4084: A9 00 0660 BOTTOM LDA #00
4086: 85 FA 0670 STA #FA ;CLEAR BIT'S NUMBER
4088: 18 0680 CLC ;CLEAR CARRY FLAG
4089: AA 0690 TAX ;INITIALIZED CLEAR STATUS
408A: A0 BE 0700 LDY #190
0710 *
408C: 98 0720 SCKBOT TYA ;CHECK END LINE
408D: 38 0730 SEC
408E: E9 05 0740 SBC #5
4090: F0 25 0750 BEQ OUT2 ;IF END GOTO OUT
4092: BD 24 40 0760 CDT2 LDA BUFF1,X ;SET STARTING ADDRESS
4095: 85 FE 0770 STA #FE
4097: BD 00 40 0780 LDA BUFF2,X
409A: 85 FF 0790 STA #FF
409C: B1 FE 0800 LDA (MEM),Y
409E: 29 7F 0810 AND #7F ;IGNORE BIT7
40A0: D0 0D 0820 BNE ST2 ;IF DATA EXISTED BOTO SAVE
40A2: EB 0830 INX ;INCREMENT BLOCK COUNTER
40A3: 8A 0840 CEN2 TXA ;CHECK END BLOCK
40A4: 38 0850 SEC
40A5: E9 24 0860 SBC #36
40A7: D0 E9 0870 BNE CDT2 ;IF NOT END GOTO CHECK DATA AGAIN
40A9: A2 00 0880 LDX #00 ;CLEAR BLOCK
40AB: 88 0890 DEY ;DECREMENT #LINE
40AC: 4C 8C 40 0900 JMP SCKBOT
40AF: 2A 0910 ST2 ROL ;CANCELLED BIT 7
40B0: E6 FA 0920 LOOP2 INC #FA
40B2: 2A 0930 ROL
40B3: 90 FB 0940 BCC LOOP2 ;IF NOT HAVE LOGIC 0 BOTO LOOP
40B5: 84 FC 0950 STY #FC ;STORE POSITION
40B7: 60 0960 OUT2 RTS
0970 *-----
0980 *
0990 *
1000 *
1010 * THIS IS PROGRAM TO FIND LEFT'S POSITION
1020 *-----
40B8: A9 00 1030 LEFT LDA #00
40BA: 85 FA 1040 STA #FA ;CLEAR BIT'S NUMBER
40BC: 18 1050 CLC ;CLEAR CARRY FLAG
40BD: AA 1060 TAX ;INITIALIZED CLEAR STATUS
40BE: A0 03 1070 LDY #5
40C0: 8A 1080 SCKLEF TXA ;CHECK END BLOCK
40C1: 38 1090 SEC
40C2: E9 24 1100 SBC #36
40C4: F0 25 1110 BEQ OUT3 ;IF END GOTO OUT
40C6: BD 24 40 1120 CDT3 LDA BUFF1,X ;SET STARTING ADDRESS
40C9: 85 FE 1130 STA #FE
40CB: BD 00 40 1140 LDA BUFF2,X
40CE: 85 FF 1150 STA #FF
40D0: B1 FE 1160 LDA (MEM),Y
40D2: 29 7F 1170 AND #7F ;IGNORE BIT7
40D4: D0 0D 1180 BNE ST3 ;IF DATA EXISTED BOTO SAVE

```



```

40D6: CB      1190      INY          ;INCREMENT LINE COUNTER
40D7: 98      1200 CEN3  TYA          ;CHECK END LINE
40D8: 38      1210      SEC
40D9: E9 BE    1220      SBC #190
40DB: D0 E9    1230      BNE CDT3      ;IF NOT END GOTO CHECK DATA AGAIN
40DD: A0 05    1240      LDY #5       ;SET STARTING
40DF: E8      1250      INX
40E0: 4C C0 40 1260      JMP SCKLEF
40E3: 2A      1270 ST3   RDL          ;CANCELLED BIT 7
40E4: E6 FA    1280 LDDP3 INC #FA
40E6: 2A      1290      ROL
40E7: 90 FB    1300      BCC LDDP3    ;IF NOT HAVE LOGIC 0 GOTO LOOP
40E9: 86 FB    1310      STX #FB
40EB: 60      1320 OUT3  RTS
1330 *-----*
1340 *
1350 *
1360 *
1370 * THIS IS PROGRAM TO FIND RIGHT'S POSITION
1380 * RIGHT-STORE SEC
1390 *-----*
40EC: A9 00    1400 RIGHT LDA #00
40EE: 85 FA    1410      STA #FA      ;CLEAR BIT'S NUMBER
40F0: 18      1420      CLC          ;CLEAR CARRY FLAG
40F1: A2 23    1430      LDX #35     ;INITIALIZED CLEAR STATUS
40F3: A0 05    1440      LDY #5
1450 *
40F5: BA      1460 SCKRIB TXA          ;CHECK END BLOCK
40F6: F0 24    1470      BEQ OUT4    ;IF END GOTO OUT
40F8: BD 24 40 1480 CDT4  LDA BUFF1,X ;SET STARTING ADDRESS
40FB: 85 FE    1490      STA #FE
40FD: BD 00 40 1500      LDA BUFF2,X
4100: 85 FF    1510      STA #FF
4102: B1 FE    1520      LDA (MEM),Y
4104: 29 7F    1530      AND #7F     ;IGNORE BIT7
4106: D0 0D    1540      BNE ST4    ;IF DATA EXISTED GOTO SAVE
4108: CB      1550      INY          ;INCREMENT LINE COUNTER
4109: 98      1560 CEN4  TYA          ;CHECK END LINE
410A: 38      1570      SEC
410B: E9 BE    1580      SBC #190
410D: D0 E9    1590      BNE CDT4    ;IF NOT END GOTO CHECK DATA AGAIN
410F: A0 05    1600      LDY #5     ;SET STARTING
4111: CA      1610      DEX
4112: 4C F3 40 1620      JMP SCKRIB
4115: E6 FA    1630 ST4   INC #FA
4117: 6A      1640      ROR
4118: 90 FB    1650      BCC ST4    ;IF NOT HAVE LOGIC 0 GOTO LOOP
411A: 86 FB    1660      STX #FB
411C: 60      1670 OUT4  RTS
00FE:        1680 MEM   .EQ #FE
4024:        1690 BUFF1 .EQ #4024
4000:        1700 BUFF2 .EQ #4000
1710      .EN

```

## SYMBOL TABLE

```

4084: BOT7DM
4024: BUFF1
4000: BUFF2
405E: CDT1
4092: CDT2
40C6: CDT3
40F8: CDT4
406F: CEN1
40A3: CEN2
40D7: CEN3
4109: CEN4
408B: LEFT
407C: LDDP1
40B0: LDDP2
40E4: LDDP3
00FE: MEM
4083: OUT1
4087: OUT2
40EB: OUT3
411C: OUT4
40EC: RIGHT
408C: SCKBOT
40C0: SCKLEF
40F5: SCKRIB
4058: SCKTOP
407B: ST1
40AF: ST2
40E3: ST3
4115: ST4
4050: TOP

```

0000 ERRORS IN ASSEMBLY

1

## MAIN

```

100 REM #SOLT store in 16705
110 HOME : HBR : HBR2 : TEXT
120 Q$ = "", B = 20: REM initial value
130 VTAB (1): INPUT "INPUT THE SLOT NUMBER "; A
140 IF A < 0 OR A > 7 THEN 160
150 POKE 237,A: GOTO 170
160 PRINT "ERROR IN SLOT NUMBER": GOTO 110
170 PRINT : PRINT : PRINT "CONTROL KEY:" : PRINT : PRINT : PRINT TAB( 8
) : "(F) FREEZE "
180 PRINT TAB( 8) : "(L) LOAD IMAGE"
190 PRINT TAB( 8) : "(S) SAVE IMAGE"
200 PRINT TAB( 8) : "(P) PRINT IMAGE"
210 PRINT TAB( 8) : "(R) RECOGNITION"
220 PRINT TAB( 8) : "(Q) QUIT"
230 D$ = CHR$( 4)
240 PRINT D$ "BLOAD BMAP2"
250 PRINT D$ "BLOAD MLD"
260 PRINT D$ "BLOAD MH12"
270 PRINT D$ "BLOAD BMAIN.N"
280 POKE 895,B: REM set #block
290 POKE 860,0: REM GET INVERSE
300 CALL 768: REM READ IN
310 CALL 25088: REM TRF
320 REM DISPLAY PICTURE
330 POKE - 16304,0: POKE - 16299,0: POKE - 16297,0
340 POKE - 16302,0
350 B = PEEK ( - 16384): IF B < 127 THEN 300: REM READ KEY
360 B$ = CHR$( B - 128)
370 IF B$ = "F" THEN 350
380 IF B$ = "P" THEN 750
390 IF B$ = "L" THEN 520
400 IF B$ = "R" THEN 2000: REM recod.
410 IF B$ = "Q" THEN TEXT : CALL - 868: END
420 IF B$ < > "S" THEN 300
430 REM SAVE
440 HOME : TEXT : VTAB (2): PRINT TAB( 15) : "**SAVE**": VTAB (4): PRINT
"SELECT DATA DRIVE: " CHR$( 8): GET E$: IF E$ > CHR$( 31) THEN PRINT
E$
450 IF E$ < > "1" AND E$ < > "2" THEN 440
460 IF E$ = "1" THEN R$ = "D1"
470 IF E$ = "2" THEN R$ = "D2"
480 VTAB (6): INPUT "ENTER NAME: "; N$
490 PRINT CHR$( 4) "BSAVE " + N$ + ",A*2000,L*1FFF," + R$
500 VTAB (20): PRINT "PRESS ANYKEY FOR RETURN TO MONITOR"
510 GET S$: BOTO 350
520 REM LOAD
530 HOME : TEXT : VTAB (2): PRINT TAB( 15) : "**LOAD**": VTAB (4): PRINT
"SELECT DATA DRIVE: " CHR$( 8): GET E$: IF E$ > CHR$( 31) THEN PRINT
E$
540 IF E$ < > "1" AND E$ < > "2" THEN 530
550 IF E$ = "1" THEN R$ = "D1"
560 IF E$ = "2" THEN R$ = "D2"
570 VTAB (6): INPUT "ENTER NAME: "; N$
580 PRINT CHR$( 4) "BLOAD " + N$ + ", " + R$
590 VTAB (20): PRINT "PRESS (P)RINT OR (D)ISPLAY"
600 VTAB (22): PRINT "PRESS ANYKEY FOR RETURN TO MONITOR"
610 B = PEEK ( - 16384): IF B < 127 THEN 610: REM READ KEY
620 B$ = CHR$( B - 128): POKE - 16368,0
630 IF B$ = "P" THEN 720
640 IF B$ < > "D" THEN 300
650 GOSUB 670: GET S$
660 TEXT : BOTO 300
670 CALL 25088: REM TRF
680 CALL 25088
690 REM DISPLAY PICTURE
700 POKE - 16304,0: POKE - 16299,0: POKE - 16297,0
710 POKE - 16302,0: RETURN
720 REM PRINT
730 REM MOVE IMAGE TO HBR1
740 GOSUB 670: REM ROTATE
750 BEB = 16384: EN = 24575: DEST = 8192
760 POKE 901,216: POKE 902,160: POKE 903,0: POKE 904,76: POKE 905,44: POKE
906,254
770 POKE 60,BEB - INT (BEB / 256) * 256: POKE 61, INT (BEB / 256)
780 POKE 62,EN - INT (EN / 256) * 256: POKE 63, INT (EN / 256)
790 POKE 66,DEST - INT (DEST / 256) * 256: POKE 67, INT (DEST / 256)
800 CALL 901
810 HOME : TEXT : VTAB (20): PRINT "PRINTING"
820 PRINT CHR$( 4) "PR#1": PRINT Q$
830 PRINT CHR$( 4) "PR#0": GET S$: TEXT
840 GOTO 300
2000 REM recod(PREP.OBJ)
2010 POKE 16705,A: POKE 16706,B
2020 TEXT : REM : CLEAR CHAIN
2030 PRINT CHR$( 4) "BRUN PREP.OBJ"
3000 B = 49200: FOR I = 1 TO 5: FOR J = 1 TO 20: C = PEEK (B): NEXT : FOR
J = 1 TO 15: NEXT : NEXT : RETURN

```

## PREP

```

100 REM !INTEGER FAX,FBX,FRZ,I,J,IL,IR,IT,JB,CR,CL,NH,J1,B1,J2,B2,EX,A
    X
110 REM 'DEFCDMMON RIX(200),LIX(200),RZX(200),LJX(200),CR,CL,NH,J1,B1,
    J2,B2,EX,AX
120 REM SEGMENTATION: detected object from background
130 REM DATA STORED IN HBR BUFFER
140 REM BGETDATA START AT #300
150 REM BBUF START AT #4000
160 REM DECISION2 START AT #4050-#4110
170 REM PROGRAM START AT #4130
180 REM I-POINT IN X-DIRECTION
190 REM J-POINT IN Y-DIRECTION
200 TEXT : HOME
210 DIM XX(100,100):D$ = CHR$(4)
220 PRINT D$"BLDAD BBUF": PRINT D$"BLDAD BGETDATA"
230 PRINT D$"BLDAD DECISION2"
240 CALL 16464: REM BOUNDARY DECISION
250 TX = PEEK(252)
260 CALL 16516:BOX = PEEK(252)
270 CALL 16568:LX = PEEK(251):PL = PEEK(250) - 1
280 LE = LX + 7
290 CALL 16620:RX = PEEK(251):MR = PEEK(250) - 1:PR = 7: REM VALUE
    EXCEED FOR SAFE
300 R = RX + 7 + PR
310 W = R - LE + 1:LL = BOX - TX: REM COUNT LAST BLDCK
320 PRINT "TOP=";TX,"LEFT=";LE: PRINT "BOTTOM=";BOX,"RIGHT";R
330 PRINT "LENGHT=";LL,"WIDTH=";W
340 J1 = 0:J2 = 0:B1 = 0:B2 = 0: REM set initial
350 REM DATA REPRESENT IN DIMENSION FORM
360 REM AC-STARTING COLUMN
370 REM AR-STARTING ROW
380 AX = LE:AY = TX
390 FOR I = 0 TO W
400 FOR J = 0 TO LL
410 REM SUB. READ DATA
420 FRX = AY + J:COL = AX + I:FBX = INT(COL / 7):FAX = COL - (FBX * 7)
    + 1
430 VTAB 15: PRINT "COL=";COL
440 POKE 250,FAX: POKE 251,FBX: POKE 252,FRX
450 CALL 768
460 VX = PEEK(253)
470 XX(I + 2,J + 2) = VX
480 NEXT J: VTAB 15: HTAB 20: PRINT "I=";I
490 NEXT I
500 REM PROJECT TO X-AXIS
510 FOR I = 0 TO W
520 CX = 0
530 FOR J = 0 TO LL
540 IF XX(I,J) < > 0 THEN CX = CX + 1:RIX(I) = CX
550 NEXT J
560 NEXT I
570 REM PROJECT TO Y-AXIS
580 FOR J = 0 TO LL
590 CY = 0
600 FOR I = 0 TO W
610 IF XX(I,J) < > 0 THEN CY = CY + 1:RJX(J) = CY
620 NEXT I
630 NEXT J
640 REM HOR. SEARCH
650 I = 0: REM INITIAL
660 IF RIX(I) < 8 THEN I = I + 1: GOTO 660
670 IF RIX(I + 1) - RIX(I) > 0 THEN I = I + 1: GOTO 670
680 IF (RIX(I + 1) - RIX(I)) > - 2 AND RIX(I + 1) - RIX(I) < 1 THEN 71
    0
690 IF RIX(I + 1) - RIX(I) > 0 THEN 710
700 I = I + 1: GOTO 680
710 IF RIX(I + 1) > 25 THEN I = I + 1: GOTO 680
720 IL = I + 1: REM EXCEEDING FOR SAFE
730 REM CONTINUE SEARCH
740 I = W
750 IF RIX(I) < 8 THEN I = I - 1: GOTO 750
760 IF RIX(I - 1) - RIX(I) > 0 THEN I = I - 1: GOTO 760
770 IF RIX(I - 1) - RIX(I) > - 2 AND RIX(I - 1) - RIX(I) < 1 THEN 800
780 IF RIX(I - 1) - RIX(I) > 0 THEN 800
790 I = I - 1: GOTO 770
800 IF RIX(I - 1) > 25 THEN I = I - 1: GOTO 770
810 IR = I - 1
820 REM VER. SEARCH
830 J = 0: REM INITIAL
840 IF RJX(J) < 8 THEN J = J + 1: GOTO 840
850 IF RJX(J + 1) - RJX(J) > 0 THEN J = J + 1: GOTO 850
860 IF RJX(J + 1) - RJX(J) < 1 AND RJX(J + 1) - RJX(J) > - 2 THEN 890
870 IF RJX(J + 1) - RJX(J) > 0 THEN 890
880 J = J + 1: GOTO 840
890 IF RJX(J + 1) > 25 THEN J = J + 1: GOTO 840
900 JT = J + 1
910 REM CONTINUE SEARCH
920 J = LL
930 IF RJX(J) < 8 THEN J = J - 1: GOTO 930
940 IF RJX(J - 1) - RJX(J) > 0 THEN J = J - 1: GOTO 940
950 IF RJX(J - 1) - RJX(J) < 1 AND RJX(J - 1) - RJX(J) > - 2 THEN 980
960 IF RJX(J - 1) - RJX(J) > 0 THEN 980
970 J = J - 1: GOTO 930
980 IF RJX(J - 1) > 25 THEN J = J - 1: GOTO 930
990 JB = J - 1
1000 PRINT "IL=";IL: PRINT "IR=";IR: PRINT "JT=";JT: PRINT "JB=";JB
1010 REM PLOTTING
1020 HGR : HCOLOR= 7:CPX = 1
1030 FOR I = IL TO IR
1040 FOR J = JT TO JB
1050 IF XX(I,J) < > CPX THEN 1070
1060 HPLLOT I,J
1070 NEXT J
1080 NEXT I
1090 REM FILTERING PROGRAM
1100 HOME : TEXT : VTAB 22: PRINT "FILTERING"
1110 TH = 5: REM SET VALUE THRESHOLD
1120 FOR I = IL TO IR

```

```

1130 FOR J = JT TO JB
1140 REM CALCULATE SUM.
1150 AA = XX(I - 1, J - 1); AB = XX(I, J - 1); AC = XX(I + 1, J - 1); AD = XX(
  I - 1, J); AE = XX(I, J); AF = XX(I + 1, J); AG = XX(I - 1, J + 1); AH = XX
  (I, J + 1); AI = XX(I + 1, J + 1)
1160 SUM = AA + AB + AC + AD + AE + AF + AG + AH + AI
1170 SUM = VAL ( RIGHT$ ( STR$ (SUM), 1))
1180 IF SUM < TH THEN VX = 0: GOTO 1210
1190 VX = 1
1200 XX(I, J) = VX * 10 + XX(I, J)
1210 NEXT J
1220 NEXT I
1230 REM PLOTTING
1240 HDR : HCOLOR = 7: CPX = 1
1250 FOR I = IL TO IR
1260 FOR J = JT TO JB
1270 IF XX(I, J) < S THEN XX(I, J) = 0: GOTO 1290
1280 H$PLOT I, J: XX(I, J) = 1
1290 NEXT J
1300 NEXT I
1310 FOR I = IL TO IR: XX(I, JT) = 0: XX(I, JB) = 0: NEXT I
1320 FOR J = JT TO JB: XX(IL, J) = 0: XX(IR, J) = 0: NEXT J: REM clear out
  of edge
1330 HOME : TEXT : V$TAB 22: PRINT "REPRESENT EDGE"
1340 REM REPRESENT OUTLINE
1350 CXZ = 0: EDX = 16: REM EDGE AS VALUE 16
1360 FOR J = JT TO JB
1370 FOR I = IL TO IR - 1
1380 IF XX(I, J) = EDX OR XX(I + 1, J) = EDX THEN 1420
1390 IF XX(I, J) = XX(I + 1, J) THEN 1420
1400 IF XX(I, J) = 0 THEN XX(I + 1, J) = EDX: CXZ = CXZ + 1: GOTO 1420
1410 XX(I, J) = EDX: CXZ = CXZ + 1
1420 NEXT I
1430 NEXT J
1440 FOR I = IL TO IR
1450 FOR J = JT TO JB - 1
1460 IF XX(I, J) = EDX OR XX(I, J + 1) = EDX THEN 1500: REM EXCEPT V-S
  ELEMENT
1470 IF XX(I, J) = XX(I, J + 1) THEN 1500
1480 IF XX(I, J) = 0 THEN XX(I, J + 1) = EDX: CXZ = CXZ + 1: GOTO 1500
1490 XX(I, J) = EDX: CXZ = CXZ + 1
1500 NEXT J
1510 NEXT I
1520 PRINT "CXZ = "; CXZ
1530 REM PLOTTING
1540 HOME : HGR : HCOLOR = 7
1550 FOR I = IL TO IR
1560 FOR J = JT TO JB
1570 IF XX(I, J) < > EDX THEN 1590
1580 H$PLOT I, J
1590 NEXT J
1600 NEXT I
1610 HOME
1620 REM FIND TOP&BOTTOM
1630 REM START TOP
1640 FOR J = JT TO JB
1650 FOR I = IL TO IR
1660 IF XX(I, J) = EDX THEN RIX(0) = I: LIX(0) = I: RJX(0) = J: LJX(0) = J:
  GOTO 1700
1670 NEXT I
1680 NEXT J
1690 BOTO 1760
1700 REM BOT.
1710 FOR J = JB TO JT STEP - 1
1720 FOR I = IL TO IR
1730 IF XX(I, J) = EDX THEN EIX = I: EJX = J: BOTO 1770
1740 NEXT I
1750 NEXT J
1760 PRINT "NO DATA EXISTED": END
1770 TEXT : PRINT "TOP = "; RJX(0), RIX(0)
1780 PRINT "BOTTOM = "; EJX, EIX
1790 TEXT
1800 REM TRACING BOUNDARY by contour tracing
1810 DTX = EDX: REM CHECK EDGE
1820 REM OUTLINE IS DEVIDED IN TWO REGIONS
1830 REM RIGHT TRACE: TRACE CLOCKWISE
1840 I = RIX(0): J = RJX(0): CR = 1: CXZ = CXZ - 1: XX(I, J) = 4: REM SET INI
  TIALIZE
1850 REM CHECK START RIGHT POINT
1860 F = 1: CP = 0: REM CP-status complete tracing
1870 GOSUB 5000
1880 IF XX(C, R) < > 0 THEN 1900
1890 F = F + 1
1900 GOSUB 5000
1910 IF XX(C, R) = 0 THEN 1960
1920 IF XX(C, R) < > DTX THEN 1980
1930 REM store position picture element
1940 XX(C, R) = 4: RIX(CR) = C: RJX(CR) = R: CR = CR + 1: CXZ = CXZ - 1: I = C
  : J = R
1950 CP = 1: GOTO 1990
1960 F = F + 1: IF F < 10 THEN 1900
1970 BOTO 1990
1980 F = F + 1: IF F < 10 THEN 1870
1990 REM check complete
2000 IF CP < > 1 THEN PRINT "text thinning": GOSUB 7000: IF CP < > 1
  THEN PRINT "ERROR": E$ = E$ + 1: GOTO 2040
2010 IF I < > EIX THEN 1860
2020 IF J < > EJX THEN 1860
2030 PRINT "COMPLETE RIGHT": PRINT "CR = "; CR: XX(EIX, EJX) = DTX: CXZ = E$
  + 1: REM BOT. POINT HAVE NOT ERASE
2040 REM LEFT TRACE: TRACE COUNTER CLOCKWISE
2050 I = LIX(0): J = LJX(0): CL = 1: REM SET INITIALIZE
2060 REM CHECK START LEFT POINT
2070 F = 1: CP = 0: REM CP-status complete tracing
2080 GOSUB 5200
2090 IF XX(C, R) < > 0 THEN 2190
2100 F = F + 1
2110 GOSUB 5200
2120 IF XX(C, R) = 0 THEN 2170
2130 IF XX(C, R) < > DTX THEN 2190
2140 REM store position picture element
2150 XX(C, R) = 4: LIX(CL) = C: LJX(CL) = R: CL = CL + 1: CXZ = CXZ - 1: I = C

```

```

IJ = R
2160 CP = 1: GOTO 2200
2170 F = F + 1: IF F < 10 THEN 2110
2180 GOTO 2200
2190 F = F + 1: IF F < 10 THEN 2080
2200 REM check complete
2210 IF CP < > 1 THEN PRINT "test thinning": GOSUB 8000: IF CP < > 1
THEN PRINT "#ERROR#":EX = EX + 1: GOTO 2250
2220 IF I < > EIX THEN 2070
2230 IF J < > EJX THEN 2070
2240 PRINT "COMPLETE LEFT"
2250 PRINT "CXX= ",CXX
2260 HBR : HCOLDR= 7
2270 FOR I = 0 TO CR - 1
2280 HPLDT RIX(I),RJX(I)
2290 NEXT I
2300 FOR I = 0 TO CL - 1
2310 HPLDT LIX(I),LJX(I)
2320 NEXT I
2330 REM CHECK HOLE
2340 IF CXX < = 3 THEN NH = 0: GOTO 2930
2350 JT = RJX(0):JB = EJX: TEXT
2360 REM FIND TOP
2370 FOR J = JT TO JB
2380 FOR I = IL TO IR
2390 IF XX(I,J) = EDX THEN I1 = I:J1 = J: GOTO 2420
2400 NEXT I
2410 NEXT J
2420 REM TRACE CLOSED CURVE
2430 I = I1:J = J1:B = 0:EIX = I1:EJX = J1
2440 F = 1:CP = 0: REM CP-status complete tracing
2450 GOSUB 5400
2460 IF XX(C,R) < > 0 THEN 2560
2470 F = F + 1
2480 GOSUB 5400
2490 IF XX(C,R) = 0 THEN 2540
2500 IF XX(C,R) < > DTX THEN 2560
2510 REM delete picture element
2520 XX(C,R) = 4:CXX = CXX - 1:I = C:J = R
2530 CP = 1: GOTO 2570
2540 F = F + 1: IF F < 10 THEN 2480
2550 GOTO 2570
2560 F = F + 1: IF F < 10 THEN 2450
2570 REM check complete
2580 IF CP < > 1 THEN PRINT "test thinning": GOSUB 9000: IF CP < > 1
THEN PRINT "#ERROR#":EX = EX + 1: GOTO 3000
2590 IF J > B THEN B = J
2600 IF I < > EIX THEN 2440
2610 IF J < > EJX THEN 2440
2620 NH = NH + 1:B1 = B
2630 PRINT "CXX= ",CXX
2640 REM CHECKHOLE#2
2650 IF CXX < = 3 THEN 2930
2660 JT = B: REM FIND TOP
2670 FOR J = JT TO JB
2680 FOR I = IL TO IR
2690 IF XX(I,J) = EDX THEN I2 = I:J2 = J: GOTO 2720
2700 NEXT I
2710 NEXT J
2720 REM TRACE HOLE#2
2730 I = I2:J = J2:B = 0:EIX = I2:EJX = J2
2740 F = 1:CP = 0: REM CP-status complete tracing
2750 GOSUB 5400
2760 IF XX(C,R) < > 0 THEN 2860
2770 F = F + 1
2780 GOSUB 5400
2790 IF XX(C,R) = 0 THEN 2840
2800 IF XX(C,R) < > DTX THEN 2860
2810 REM delete picture element
2820 XX(C,R) = 4:CXX = CXX - 1:I = C:J = R
2830 CP = 1: GOTO 2870
2840 F = F + 1: IF F < 10 THEN 2780
2850 GOTO 2870
2860 F = F + 1: IF F < 10 THEN 2750
2870 REM check complete
2880 IF CP < > 1 THEN PRINT "test thinning": GOSUB 9000: IF CP < > 1
THEN PRINT "#ERROR#":EX = EX + 1: GOTO 3000
2890 IF J > B THEN B = J
2900 IF I < > EIX THEN 2740
2910 IF J < > EJX THEN 2740
2920 NH = NH + 1:B2 = B
2930 PRINT "CXX= ",CXX:H1 = (J1 + B1) / 2:H2 = (J2 + B2) / 2
2940 PRINT "HOLE#1 ON ",H1
2950 PRINT "HOLE#2 ON ",H2
2960 IF EX > 0 THEN PRINT "TOTAL ERROR= ",EX
2970 PRINT "COMPLETE ": PRINT "# HOLE= ",NH
2980 PRINT "J1= ",J1,"B1= ",B1
2990 PRINT "J2= ",J2,"B2= ",B2
3000 REM ! CLEAR CHAIN
3010 PRINT CHR# (4)"BRUN REC.OBJ"
3000 REM TEST CONNECTED
3010 REM INPUT F-FREE MAN DIRECTION
3030 REM OUTPUT DI-DIRECTION
3040 REM R,C-RDN&COLUMN
3050 DN F GOTO 5130,5135,5060,5070,5090,5100,5110,5120,5130
3060 C = I + 1:R = J:DI = 0: GOTO 5140
3070 C = I + 1:R = J + 1:DI = 1: GOTO 5140
3090 C = I:R = J + 1:DI = 2: GOTO 5140
3100 C = I - 1:R = J + 1:DI = 3: GOTO 5140
3110 C = I - 1:R = J:DI = 4: GOTO 5140
3120 C = I - 1:R = J - 1:DI = 5: GOTO 5140
3130 C = I:R = J - 1:DI = 6: GOTO 5140
3135 C = I + 1:R = J - 1:DI = 7
5140 RETURN
5200 DN F GOTO 5130,5120,5110,5100,5090,5070,5060,5135,5130: REM trace
counter clockwise
3400 DN F GOTO 5090,5100,5110,5120,5130,5135,5060,5070,5090: REM trace
counter clockwise
7000 REM test case thin character
7010 F = 1:CP = 0: REM CP-status complete tracing
7020 GOSUB 5000
7030 IF XX(C,R) < > 0 THEN 7130

```

```

7040 F = F + 1
7050 GOSUB 5000
7060 IF XX(C,R) = 0 THEN 7110
7070 IF XX(C,R) < > 4 THEN 7130
7080 REM store position picture element
7090 RIZ(CR) = C:RIZ(CR) = R:CR = CR + 1:I = C:J = R
7100 CP = 1: GOTO 7140
7110 F = F + 1: IF F < 10 THEN 7050
7120 GOTO 7140
7130 F = F + 1: IF F < 10 THEN 7020
7140 RETURN
8000 REM test case thin character (left)
8010 F = 1:CP = 0: REM CP-status complete tracing
8020 GOSUB 5200
8030 IF XX(C,R) < > 0 THEN 8130
8040 F = F + 1
8050 GOSUB 5200
8060 IF X2(C,R) = 0 THEN 8150
8070 IF X2(C,R) < > 4 THEN 8130
8080 REM store position picture element
8090 LIX(CL) = C:LIX(CL) = R:CL = CL + 1:I = C:J = R
8100 CP = 1: GOTO 8140
8110 F = F + 1: IF F < 10 THEN 8050
8120 GOTO 8140
8130 F = F + 1: IF F < 10 THEN 8020
8140 RETURN
9000 REM test case thin character (close curve)
9010 F = 1:CP = 0: REM CP-status complete tracing
9020 GOSUB 5400
9030 IF XX(C,R) < > 0 THEN 9130
9040 F = F + 1
9050 GOSUB 5400
9060 IF XX(C,R) = 0 THEN 9110
9070 IF XX(C,R) < > 4 THEN 9130
9080 REM store position picture element
9090 I = C:J = R
9100 CP = 1: GOTO 9140
9110 F = F + 1: IF F < 10 THEN 9050
9120 GOTO 9140
9130 F = F + 1: IF F < 10 THEN 9020
9140 RETURN

```

```

100 REM !INTEGER FAX,FBX,FRX,I,J,IL,IR,JT,JB,CR,CL,U1,UB,UC,UD,UE,EX,A
    Z
110 REM !USECOMMON RIX(200),LIX(200),RJX(200),LJX(200),CR,CL,U1,UB,UC,U
    D,UE,EX,AX
120 REM Split Algorithm
130 REM E(1)-ERROR ON EACH SEGMENT
150 REM XR/LX(1)-STORE POSITION X CORNER
160 REM YR/LX(1)-STORE POSITION Y CORNER
170 REM AR/L(1)-ANGLE AT CORNER I
180 REM H1X,H2X-COORD. X,Y HEAD OF RIGHT SIDE
190 REM H3X,H4X-COORD. X,Y HEAD OF LEFT SIDE
200 REM T1X,T2X-COORD. X,Y TAIL OF RIGHT SIDE
210 REM T3X,T4X-COORD. X,Y TAIL OF LEFT SIDE
230 DIM H1X(40),H2X(40),H3X(40),H4X(40),T1X(40),T2X(40),T3X(40),T4X(40)
    ,IRX(40),YRX(40),XLX(40),YLY(40),AR(40),AL(40),LPX(40),ULX(40),RPX(
    40),URX(40),E(40),DL(40),DR(40),FL(40),FR(40),D(40),O(40),BRX(4),B
    LX(4),ZNX(10),A*(5),B*(5)
235 DIM BIX(5),SLX(3),SRX(3),ELX(3),ERZ(3)
240 PI = 3.1416/8M = 10;ML = B;MR = ML
290 SP = 0: REM SET PARAMETER
300 REM PROGRAM FOR ARRANGE SEGMENT B FORMAT
310 REM STORE IN STARTING #4150
320 REM DETAILS FOR EACH SEGMENT HAVE 2 FIELDS
330 REM LPX(I)&RPX(I)-STARTPOINTS OF THE SEGMENTS
340 REM ULX(I)&URX(I)-LENGTH OF EACH SEGMENT
350 REM NP-NUMBER PIXELS PER SEGMENT
360 REM SP-START POINT FOR EACH SEGMENTS
370 REM I-SEGMENTS' ORDER
380 REM C-TEMPORARY COUNTER
390 REM LESTHAND FIRST
400 PRINT "CR=";CR: PRINT "CL=";CL
410 NP = INT (CL / ML):E = CL - NP * ML
420 IF (NP - E) > 0 THEN 440
430 ML = ML + 1:F = E: GOTO 450
440 F = NP + E: REM FINAL SEGMENT
450 FOR I = 1 TO ML - 1
460 LPX(I) = SP:ULX(I) = NP
470 SP = SP + NP: REM UPDATE
480 NEXT I: REM LOOPING
490 I = ML:LPX(I) = SP:ULX(I) = F
500 PRINT "NUMBER SEGMENT:";ML
510 REM STARTING
520 REM CALCULATED ALL E1
530 FOR I = 1 TO ML
540 SP = LPX(I):NP = ULX(I)
550 EP = SP + NP - 1
570 GOSUB 25000: REM CALL E1
590 E(I) = E2:DL(I) = D0:FL(I) = FE
600 NEXT I
610 S = 0: REM BEFORE SPLIT
620 PRINT : PRINT "Check Split": PRINT
630 DM = ML: REM ML-variable
640 REM Check Split*****
650 FOR I = 1 TO DM
660 IF E(I) < = 8M THEN 680: REM NEXT
670 II = I: I = DM: S = S + 1: GOSUB 23500: REM Split
680 NEXT I
690 IF ML > DM THEN 630: REM Split continue
700 PRINT "NUMBER OF SPLIT=";S
710 REM Check Merge*****
720 REM MERGE 1 pair per 1 time
730 PRINT "Test Condition Merge"
740 ME = 0: REM BEFORE MERGE
750 N = 0: REM Initial
760 FOR I = 1 TO ML - 1
770 REM SUBM
780 SP = LPX(I):EP = LPX(I + 1) + ULX(I + 1) - 1
790 NP = EP - SP + 1
810 GOSUB 25000: REM CAL. E1
820 IF E2 > 8M THEN 860
830 REM store in stack
840 N = N + 1:ME = ME + 1
850 ST(N) = E2:O(N) = 1
860 NEXT I
870 PRINT "NUMBER OF NEW SEGMENT(E1 < Emax) = ";E
880 IF N = 0 THEN 1190: REM Out
890 REM sorting decreation
900 REM Sorting
910 REM INPUT N,ST(J)
920 D = 2 ^ INT (LOG (N) / LOG (2)) - 1
930 FOR I = 1 TO N - D
940 FOR J = I TO I STEP - D
950 IF ST(J) < = ST(J + D) THEN 1000
960 TT = ST(J):OT = O(J)
970 ST(J) = ST(J + D):O(J) = O(J + D)
980 ST(J + D) = TT:O(J + D) = OT
990 NEXT J
1000 NEXT I
1010 D = INT (D / 2)
1020 IF D > 0 THEN 930
1080 E1 = ST(1): REM pick first element
1090 S1 = 0(1):LS = ULX(S1) + ULX(S1 + 1)
1100 REM Update Parameter
1110 FOR I = S1 + 1 TO ML - 1 STEP 1
1120 LPX(I) = LPX(I + 1):ULX(I) = ULX(I + 1)
1130 E(I) = E(I + 1):DL(I) = DL(I + 1):FL(I) = FL(I + 1)
1140 NEXT I
1150 REM Calculate New Segment
1160 SP = LPX(S1):EP = SP + LS - 1:NP = LS
1170 GOSUB 25000:ULX(S1) = DL(S1) = D0:FL(S1) = FE:E(S1) = E2:ML = M
    L - 1
1180 IF ML < > 1 THEN 740
1190 REM START RIGHT
1200 V1 = 10 * PI / 180
1210 SP = 0: REM SET PARAMETER
1220 NP = INT (CR / MR):E = CR - NP * MR
1230 IF (NP - E) > 0 THEN 1250
1240 MR = MR + 1:F = E: GOTO 1240
1250 F = NP + E: REM FINAL SEGMENT

```

```

1260 FOR I = 1 TO MR - 1
1270 RPX(I) = SP:URX(I) = NP
1280 SP = SP + NP: REM UPDATE
1290 NEXT I: REM LOOPING
1300 I = MR:RPX(I) = SP:URX(I) = F
1310 PRINT "NUMBER SEGMENT:";MR
1320 REM STARTING
1330 REM CALCULATED ALL E1
1340 FOR I = 1 TO MR
1350 SP = RPX(I):NP = URX(I)
1360 EP = SP + NP - 1
1380 GOSUB 26000: REM CALL E1
1400 E(I) = E2:DR(I) = DD:FR(I) = FE
1410 NEXT I
1420 S = 0: REM BEFORE SPLIT
1430 PRINT: PRINT "Check Split": PRINT
1440 OM = MR: REM MR-variable
1450 REM Check Split*****
1460 FOR I = 1 TO OM
1470 IF E(I) < OM THEN 1490: REM NEXT
1480 I1 = I: OM = S + 1: GOSUB 24000: REM Split
1490 NEXT I
1500 IF MR > OM THEN 1440: REM Split continue
1510 PRINT "NUMBER OF SPLIT=";S
1520 REM Check Merge*****
1530 REM MERGE 1 pair per 1 time
1540 PRINT "Test Condition Merge"
1550 ME = 0: REM BEFORE MERGE
1560 N = 0: REM Initial
1570 FOR I = 1 TO MR - 1
1580 REM SUBM
1590 SP = RPX(I):EP = RPX(I + 1) + URX(I + 1) - 1
1600 NP = EP - SP + 1
1620 GOSUB 26000: REM CAL. E1
1630 IF E2 > OM THEN 1670
1640 REM store in stack
1650 N = N + 1:ME = ME + 1
1660 ST(N) = E2:D(N) = I
1670 NEXT I
1680 PRINT "NUMBER OF NEW SEGMENT(E1 < Emax) = ";ME
1690 IF N = 0 THEN 2000: REM Out
1700 REM sorting decreation
1710 REM Sorting
1720 REM INPUT N,ST(I)
1730 D = 2 ^ INT ( LOG (N) / LOG (2) ) - 1
1740 FOR I = 1 TO N - D
1750 FOR J = I TO I STEP - D
1760 IF ST(J) < ST(J + D) THEN 1810
1770 TT = ST(J):DT = D(J)
1780 ST(J) = ST(J + D):D(J) = D(J + D)
1790 ST(J + D) = TT:D(J + D) = DT
1800 NEXT J
1810 NEXT I
1820 D = INT (D / 2)
1830 IF D > 0 THEN 1740
1890 E1 = ST(1): REM pick first element
1900 S1 = 0(1):LS = URX(S1) + URX(S1 + 1)
1910 REM Update Parameter
1920 FOR I = S1 + 1 TO MR - 1 STEP 1
1930 RPX(I) = RPX(I + 1):URX(I) = URX(I + 1)
1940 E(I) = E(I + 1):DR(I) = DR(I + 1):FR(I) = FR(I + 1)
1950 NEXT I
1960 REM Calculate New Segment
1970 SP = RPX(S1):EP = SP + LS - 1:NP = LS
1980 GOSUB 26000:URX(S1) = LS:DR(S1) = DD:FR(S1) = FE:E(S1) = E2:MR = M
R = 1
1990 IF MR < > 1 THEN 1950
2000 REM ending split&merge
2010 REM CAL.LENGTH(RIGHT)
2020 REM H1X(I),H2X(I)-HEAD SEGMENT (2 digit after point)
2030 REM T1X(I),T2X(I)-TAIL SEGMENT
2040 PRINT "***RIGHT SIDE***"
2050 FOR I = 1 TO MR
2060 SP = RPX(I):NP = URX(I):EP = SP + NP - 1:X1 = RIX(SP):X2 = RIX(EP)
Y1 = 120 - RJX(SP):Y2 = 120 - RJX(EP)
2070 DD = DR(I):FE = FR(I)
2080 DET = - 1
2090 C1 = Y1 * SIN (FE) - X1 * COS (FE)
2100 H1X(I) = INT (100 * (C1 * COS (FE) - DD * SIN (FE)) / DET)
2110 H2X(I) = INT (100 * (- DD * COS (FE) - C1 * SIN (FE)) / DET)
2120 C2 = Y2 * SIN (FE) - X2 * COS (FE)
2130 T1X(I) = INT (100 * (C2 * COS (FE) - DD * SIN (FE)) / DET)
2140 T2X(I) = INT (100 * (- DD * COS (FE) - C2 * SIN (FE)) / DET)
2170 NEXT I
2180 REM CAL. LENGTH(LEFT)
2190 REM H3X(I),H4X(I)-HEAD SEGMENT(2 digit after point)
2200 REM T3X(I),T4X(I)-TAIL SEGMENT
2210 PRINT "***LEFT SIDE***"
2220 FOR I = 1 TO ML
2230 SP = LPX(I):NP = ULX(I):EP = SP + NP - 1:X1 = LIX(SP):X2 = LIX(EP)
Y1 = 120 - LJX(SP):Y2 = 120 - LJX(EP)
2240 DD = DL(I):FE = FL(I)
2250 DET = - 1
2260 C1 = Y1 * SIN (FE) - X1 * COS (FE)
2270 H3X(I) = INT (100 * (C1 * COS (FE) - DD * SIN (FE)) / DET)
2280 H4X(I) = INT (100 * (- DD * COS (FE) - C1 * SIN (FE)) / DET)
2290 C2 = Y2 * SIN (FE) - X2 * COS (FE)
2300 T3X(I) = INT (100 * (C2 * COS (FE) - DD * SIN (FE)) / DET)
2310 T4X(I) = INT (100 * (- DD * COS (FE) - C2 * SIN (FE)) / DET)
2340 NEXT I
2350 REM PLOT
2360 HGR 1:HCOLOR= 7
2370 HPLLOT 0,0 TO 0,120 TO 100,120
2380 HPLLOT 160,0 TO 160,120 TO 260,120
2390 FOR I = 0 TO CR - 1
2400 HPLDT 180 + RIX(I),RJX(I)
2410 NEXT I
2420 FOR I = 0 TO CL - 1
2430 HPLDT 180 + LIX(I),LJX(I)
2440 NEXT I
2450 PRINT "Total right Segment:";MR
2460 PRINT "Total left Segment:";ML

```



```

2470 REM FIND BREAK POINT
2480 XT = INT ((H1X(I) + H3X(I)) / 200); YT = INT ((H2X(I) + H4X(I)) /
200)
2490 IF MR = 1 THEN 2530
2500 FOR I = MR - 1 TO 1 STEP - 1
2510 XRX(I) = INT ((T1X(I) + H1X(I + 1)) / 200); YRX(I) = INT ((T2X(I) +
H2X(I + 1)) / 200)
2520 NEXT I
2530 XB = INT ((T1X(MR) + T3X(ML)) / 200); YB = INT ((T2X(MR) + T4X(ML)
) / 200)
2540 IF ML = 1 THEN 2580
2550 FOR I = 1 TO ML - 1
2560 XLX(I) = INT ((T3X(I) + H3X(I + 1)) / 200); YLX(I) = INT ((T4X(I) +
H4X(I + 1)) / 200)
2570 NEXT I
2580 REM angle break point
2590 REM input-XT,YT,XRX(I),YRX(I),XB,YB,XLX(I),YLX(I)
2600 REM left
2610 IF ML = 1 THEN 2750
2620 DX = XLX(I) - XT; DY = YLX(I) - YT
2630 SOBUB 27000; FL(I) = AB
2640 DX = XB - XLX(ML - 1); DY = YB - YLX(ML - 1)
2650 SOBUB 27000; FL(ML) = AB
2655 IF ML = 2 THEN 2700
2660 FOR I = 2 TO ML - 1
2670 DX = YLX(I) - YLX(I - 1); DY = XLX(I) - XLX(I - 1)
2680 SOBUB 27000; FL(I) = AB
2690 NEXT I
2700 FOR I = 1 TO ML - 1
2710 AL(I) = PI + FL(I) - FL(I + 1)
2720 IF AL(I) < 0 THEN AL(I) = 2 * PI + AL(I)
2730 IF AL(I) > 2 * PI THEN AL(I) = AL(I) - 2 * PI
2740 NEXT I
2750 REM right
2760 IF MR = 1 THEN 2920
2770 BY = YT - YRX(I); DX = XT - XRX(I)
2780 SOBUB 27000; FR(I) = AB
2790 BY = YRX(MR - 1) - YB; DX = XRX(MR - 1) - XB
2800 SOBUB 27000; FR(MR) = AB
2805 IF MR = 2 THEN 2850
2810 FOR I = MR - 1 TO 2 STEP - 1
2820 BY = YRX(I - 1) - YRX(I); DX = XRX(I - 1) - XRX(I)
2830 SOBUB 27000; FR(I) = AB
2840 NEXT I
2850 FOR I = MR - 1 TO 1 STEP - 1
2860 AR(I) = PI + FR(I + 1) - FR(I)
2870 IF AR(I) = 2 * PI THEN AR(I) = 0
2880 IF AR(I) = 0 THEN AR(I) = 2 * PI
2890 IF AR(I) < 0 THEN AR(I) = 2 * PI + AR(I)
2900 IF AR(I) > 2 * PI THEN AR(I) = AR(I) - 2 * PI
2910 NEXT I
2920 REM PLOT POLYGONAL SEGMENT
2930 HPLLOT XT,120 - YT
2940 IF MR = 1 THEN 2980
2950 FOR I = 1 TO MR - 1
2960 HPLLOT TO XRX(I),120 - YRX(I)
2970 NEXT I
2980 HPLLOT TO XB,120 - YB
2990 IF ML = 1 THEN 3030
3000 FOR I = ML - 1 TO 1 STEP - 1
3010 HPLLOT TO XLX(I),120 - YLX(I)
3020 NEXT I
3030 HPLLOT TO XT,120 - YT
3040 PRINT "Total right Segments:";MR
3050 PRINT "Total left Segments:";ML
3190 PRINT "reduce size "
3200 FOR I = 1 TO MR
3210 H1X(I) = INT (H1X(I) / 100); T1X(I) = INT (T1X(I) / 100); H2X(I) =
INT (H2X(I) / 100); T2X(I) = INT (T2X(I) / 100)
3220 FR(I) = FR(I) * 180 / PI; AR(I) = AR(I) * 180 / PI
3230 NEXT I
3240 FOR I = 1 TO ML
3250 H3X(I) = INT (H3X(I) / 100); T3X(I) = INT (T3X(I) / 100); H4X(I) =
INT (H4X(I) / 100); T4X(I) = INT (T4X(I) / 100)
3260 FL(I) = FL(I) * 180 / PI; AL(I) = AL(I) * 180 / PI
3270 NEXT I
3370 UB = 120 - UB; UC = 120 - UC; UD = 120 - UD; UE = 120 - UE; PRINT "J1=
";UB,"B1= ";UC
3380 PRINT "J2= ";UD,"B2= ";UE
3390 Y = YT - YB; U3 = 0
3410 REM feature generation
4200 PRINT "U1= ";U1
4210 REM U2-hole description variable
4220 REM U2=1,if center is in zone 6 or 7 and h1 in zone 3,6,7(low ho
le)
4230 REM U2=2,if center is in zone 1 and h2 in zone 3,4,5(high hole)
4240 REM U2=0,otherwise
4250 IF U1 = 0 OR U1 > .1 THEN U2 = - 1; GOTO 4290
4260 U2 = 0; C1 = (UB + UC) / 2; C1 = C1 - YB; REM initial case
4270 IF UB - YB < 3 * Y / 5 AND C1 < 2 * Y / 5 THEN U2 = 1
4280 IF UC - YB > 2 * Y / 5 AND C1 > Y / 2 THEN U2 = 2
4290 PRINT "U2= ";U2
4300 REM U3-over segment /fs=90/<10 degree
4310 REM FR=R1X(160+I);H1=R1X(I);T1=R1X(140+I)
4320 REM H2=R1X(80+I);T2=R1X(120+I)
4330 FOR I = 1 TO MR
4340 IF ABS (FR(I) - 90) > 10 THEN 4380
4350 LT = SQR (((H1X(I) - T1X(I)) ^ 2) + ((H2X(I) - T2X(I)) ^ 2))
4360 IF LT < Y / 3 THEN 4380
4370 U3 = U3 + 1; PRINT "right segments ";I; PRINT "HEAD X,Y: ";H1X(I),"
";H2X(I); PRINT "TAIL X,Y: ";T1X(I)," ";T2X(I); PRINT "SLOPE=";FR
(I)
4380 NEXT I
4390 FOR I = 1 TO ML
4400 IF ABS (FL(I) - 90) > 10 THEN 4440
4410 LT = SQR (((H3X(I) - T3X(I)) ^ 2) + ((H4X(I) - T4X(I)) ^ 2))
4420 IF LT < Y / 3 THEN 4440
4430 U3 = U3 + 1; PRINT "left segments ";I; PRINT "HEAD X,Y: ";H3X(I),"
";H4X(I); PRINT "TAIL X,Y: ";T3X(I)," ";T4X(I); PRINT "SLOPE=";FL(I)
4440 NEXT I

```

```

4450 PRINT "U3= "U3
4460 REM U4=concave on left side(angle<10)
4470 I = 1:U4 = 0
4480 IF ML = 1 THEN 4670: REM out
4490 IF AL(I) > = 160 THEN 4540: REM CONVEX
4500 U4 = U4 + 1
4510 B*(U4) = B*(U4) + BTR*(I)
4520 I = I + 1: IF I > ML - 1 THEN 4560
4530 IF AL(I) < 160 THEN 4510
4540 I = I + 1
4550 IF I < ML THEN 4490
4560 IF U4 = 0 THEN 4670
4570 FOR I = 1 TO U4
4580 B = VAL ( LEFT* (B*(I),1)):E = VAL ( RIGHTS (B*(I),1))
4590 IF B = E THEN B = YLX(B): BOTO 4650
4600 B = 120
4610 FOR IC = B TO E
4620 IF YLX(IC) > B THEN 4640
4630 B = YLX(IC)
4640 NEXT IC
4650 BLX(I) = B
4660 NEXT I
4670 REM and left
4680 REM U5=concave on right side
4690 I = 1:U5 = 0
4700 IF MR = 1 THEN 4890: REM out
4710 IF AR(I) > = 160 THEN 4760: REM CONVEX
4720 U5 = U5 + 1
4730 A*(U5) = A*(U5) + BTR*(I)
4740 I = I + 1: IF I > MR - 1 THEN 4780
4750 IF AR(I) < 160 THEN 4730
4760 I = I + 1
4770 IF I < MR THEN 4710
4780 IF U5 = 0 THEN 4890
4790 FOR I = 1 TO U5
4800 B = VAL ( LEFT* (A*(I),1)):E = VAL ( RIGHTS (A*(I),1))
4810 IF B = E THEN B = YRX(B): BOTO 4870
4820 B = 120
4830 FOR IC = B TO E
4840 IF YRX(IC) > B THEN 4860
4850 B = YRX(IC)
4860 NEXT IC
4870 BRX(I) = B
4880 NEXT I
4890 REM and right
4900 REM U6,U7,U8-distance from bot. of char. the first,second,and thi
rd concave arc
4910 IF U4 = 0 THEN I = 0: BOTO 5000
4920 FOR I = 1 TO U4
4930 BLX(I) = BLX(I) - YB
4940 IF BLX(I) < Y / 3 THEN ZNX(I) = 7: BOTO 4990
4950 IF BLX(I) < 2 * Y / 5 THEN ZNX(I) = 6: BOTO 4990
4960 IF BLX(I) < 3 * Y / 5 THEN ZNX(I) = 5: BOTO 4990
4970 IF BLX(I) < 4 * Y / 5 THEN ZNX(I) = 4: BOTO 4990
4980 IF BLX(I) < Y THEN ZNX(I) = 3: BOTO 4990
4990 NEXT I: I = I - 1
5000 IF U5 = 0 THEN 5090
5010 FOR J = U5 TO 1 STEP - 1
5020 BRX(J) = BRX(J) - YB
5030 IF BRX(J) < Y / 3 THEN I = I + 1:ZNX(I) = 7: BOTO 5080
5040 IF BRX(J) < 2 * Y / 5 THEN I = I + 1:ZNX(I) = 6: BOTO 5080
5050 IF BRX(J) < 3 * Y / 5 THEN I = I + 1:ZNX(I) = 5: BOTO 5080
5060 IF BRX(J) < 4 * Y / 5 THEN I = I + 1:ZNX(I) = 4: BOTO 5080
5070 IF BRX(J) < Y THEN I = I + 1:ZNX(I) = 3: BOTO 5080
5080 NEXT J
5090 PRINT "U4= "U4,"U5= "U5
5100 U6 = ZNX(1):U7 = ZNX(2):U8 = ZNX(3)
5110 PRINT "U6= "U6,"U7= "U7,"U8= "U8
5120 REM U11,U12,U13,U14(U8,UC,UD,UE)
5130 IF U1 = 0 THEN 5200
5140 VV = UB - YB: B0SUB 35000:UB = ZN: PRINT "U11= "U1
5150 VV = UC - YB: B0SUB 35000:UC = ZN: PRINT "U12= "U1
5160 IF U1 = 1 THEN 5200
5170 VV = UD - YB: B0SUB 35000:UD = ZN: PRINT "U13= "U1
5180 VV = UE - YB: B0SUB 35000:UE = ZN: PRINT "U14= "U1
5190 REM V4-direction seq. followed right concave.
5200 IF U5 = 0 THEN V4 = - 1: BOTO 5230
5210 BR = VAL ( RIGHTS* (A*(U5),1)) + 1
5220 AB = FR(BR): B0SUB 36000:V4 = BC
5230 PRINT "V4= "V4
5240 REM V3-direction seq. followed left concave.
5250 IF U4 = 0 THEN V3 = - 1: BOTO 5280
5260 EL = VAL ( RIGHTS* (B*(U4),1)) + 1: PRINT "EL(V3)= "EL
5270 AB = FL(EL): PRINT "AB(V3)= "AB: B0SUB 36000:V3 = BC
5280 PRINT "V3= "V3
5290 REM V5-direction of seq. before i's left concave
5300 IF U4 = 0 THEN V5 = - 1: BOTO 5330
5310 B = VAL ( LEFT* (B*(I),1)):AB = FL(B): PRINT "B(V5)= "B,"AB= "AB
: B0SUB 36000
5320 V5 = BC
5330 PRINT "V5= "V5
5340 REM U15,U16-bisecton code of first on left and last on right ARC(
UF,V1)
5350 X = 0:XC = 0: REM initial
5360 IF U4 = 0 THEN 5330: REM out
5370 FOR I = 1 TO U4
5380 B = VAL ( LEFT* (B*(I),1)):E = VAL ( RIGHTS* (B*(I),1))
5390 IF B = E THEN 5320
5400 BLX(I) = B:ELX(I) = E
5410 IF FL(B) > 180 THEN FL(E + 1) = FL(E + 1) + 180: BOTO 5430
5420 FL(E + 1) = FL(E + 1) - 180
5430 DF = ABS (FL(B) - FL(E + 1))
5440 IF DF < 180 THEN 5470
5450 DF = 360 - DF: IF FL(B) > FL(E + 1) THEN AB = FL(B) + DF / 2: BOTO
5490
5460 AB = FL(E + 1) + DF / 2: BOTO 5490
5470 IF FL(B) > FL(E + 1) THEN AB = FL(E + 1) + DF / 2: BOTO 5490
5480 AB = FL(B) + DF / 2
5490 IF AB > 360 THEN AB = AB - 360
5500 PRINT "AB= "AB: B0SUB 36000
5510 X = I + 1:XC = XC + 1:BX(X) = BC
5520 NEXT I

```



```

5530 IF X = 0 THEN UF = - 1: GOTO 5550
5540 UF = BIX(1)
5550 REM start right
5560 K = 0: REM initial
5570 IF U5 = 0 THEN 5740: REM.out
5580 FOR I = U5 TO 1 STEP - 1
5590 E = VAL ( LEFT$ (A*(I),1)):S = VAL ( RIGHT$ (A*(I),1))
5600 IF S = E THEN 5730
5610 SRX(I) = S:ERX(I) = E
5620 IF FR(S + 1) > 180 THEN FR(E) = FR(E) + 180: GOTO 5640
5630 FR(E) = FR(E) - 180
5640 DF = ABS (FR(S + 1) - FR(E))
5650 IF DF < 180 THEN 5680
5660 DF = 360 - DF: IF FR(S + 1) > FR(E) THEN AB = FR(S + 1) + DF / 2: GOTO
5700
5670 AB = FR(E) + DF / 2: GOTO 5700
5680 IF FR(S + 1) > FR(E) THEN AG = FR(E) + DF / 2: GOTO 5700
5690 AG = FR(S + 1) + DF / 2
5700 IF AG > 360 THEN AG = AG - 360
5710 PRINT "AB= "AB: B0SUB 36000
5720 X = X + 1:XC = XC + 1:BIX(X) = BC
5730 NEXT I
5740 IF X = 0 THEN V1 = - 1: GOTO 5760
5750 V1 = BIX(X)
5760 PRINT "U15= "U15,"U16= "V1
5770 REM U17-number of concave ARC
5780 V2 = XC: PRINT "U17= "V2
5790 REM U9-number of corner in left
5970 REM U10-number of corner in right(UA)
5980 U9 = ELX(1) - BLX(1) + 1:UA = SRX(X) - ERX(X) + 1
5990 PRINT "U9= "U9,"U10= "UA
6000 REM classification tree
6010 IF U1 = 0 THEN 6250
6020 IF U1 = 1 THEN 6040
6030 IF U1 = 2 THEN 6220
6040 REM case 1 hole
6050 IF U4 + U5 < = 2 THEN 6120
6060 IF U2 = 0 THEN 6940
6070 IF U2 = 2 THEN 6100
6080 IF (UF = 4 OR UF = 5 OR UB = 4) OR (V1 = 4 OR V1 = 5) THEN 6980: REM
6
6090 GOTO 6920: REM 2
6100 IF UC = 3 OR UC = 4 THEN 6920: REM 2
6110 GOTO 6940: REM 4
6120 IF U2 = 0 THEN 6190: REM 0&4&6
6130 IF U2 = 1 THEN 6170: REM 6&8
6140 IF U5 = 0 THEN 6990: REM 7
6150 IF (V4 = 3 OR V4 = 4 OR V4 = 5) AND (U7 < > 3 OR U7 < > 4) THEN
6920: REM 2
6160 GOTO 6990: REM 9
6170 IF U5 = 2 THEN 6980: REM 8
6180 GOTO 6960: REM 6
6190 IF U6 = 3 OR U6 = 4 THEN 6215: REM 0&6
6200 IF U4 + U5 = 0 THEN 6900: REM 0
6210 GOTO 6940
6215 IF U4 = 0 THEN 6960
6217 GOTO 6900
6220 REM case 2 hole(2,8)
6230 IF UB < > 3 AND U6 < > 3 AND U4 + U5 > = 3 THEN 6920: REM 2..
6240 GOTO 6980
6250 REM case no hole
6260 IF U4 + U5 = 0 THEN 6910: REM 1
6270 IF U4 + U5 = 1 THEN 6330: REM 1concave
6280 IF U4 + U5 = 2 THEN 6400: REM 2concave
6290 IF U4 + U5 = 3 THEN 6680: REM 3concave
6300 REM U4+U5>=4
6305 IF U6 = 3 THEN 6930: REM 3
6310 IF (UF = 0 OR UF = 1 OR UF = 2) AND V1 = - 1 THEN 6970: REM 7
6320 GOTO 6940: REM 4
6330 REM case 1concave
6340 IF U4 = 0 THEN 6960: REM 6
6345 IF V2 < > 0 THEN 6970
6350 IF U3 > = 1 THEN 6910
6360 GOTO 6970
6400 REM case 2concave
6410 IF U4 = 2 THEN 6560: REM 1&3&4&7
6420 IF U7 = 3 THEN 6470: REM 5&7
6430 IF V2 < > 0 THEN 6510
6440 IF U6 = 6 OR U6 = 7 THEN 6660: REM 1&5
6450 IF V4 = 1 OR V4 = 2 THEN 6970
6460 GOTO 6920: REM 2
6470 IF U6 < > 3 OR U6 < > 4 THEN 6950: REM 5
6480 GOTO 6970
6490 IF V3 = 4 THEN 6910
6500 IF V3 = 5 THEN 6920
6505 GOTO 6950
6510 IF U7 = 6 OR U7 = 7 THEN 6600
6520 IF V4 = 3 OR V4 = 4 OR V4 = 5 THEN 6860: REM 3&9
6530 IF V4 = 1 OR V4 = 2 THEN 6970
6540 IF UA > = 3 THEN 6990: REM 9
6550 GOTO 6940
6560 IF UF = 5 OR UF = 6 THEN 6940
6570 IF V5 = 5 OR V5 = 6 THEN 6595: REM 1&7
6580 IF U7 = 7 OR V3 = 4 OR V3 = 5 THEN 6930
6590 GOTO 6970: REM 7
6595 IF U3 > = 1 THEN 6910
6597 GOTO 6970
6600 IF V1 = 5 OR V1 = 6 THEN 6640: REM 2&4
6610 IF U6 = 6 OR U6 = 7 THEN 6490: REM 1&2&5
6620 IF V4 = 1 OR V4 = 2 THEN 6970
6630 GOTO 6920: REM 2
6640 IF U7 = 7 OR U7 = 6 THEN 6920
6650 GOTO 6940
6660 IF U3 > = 1 AND (U7 = 6 OR U7 = 7) THEN 6910: REM 1
6670 IF U7 = 6 OR U7 = 7 THEN 6920
6675 GOTO 6950
6680 IF UB = 3 THEN 6790
6690 IF U4 = 1 THEN 6840
6700 IF ABS (U7 - U6) > 1 THEN 6760: REM 2&3&5
6710 IF V2 > = 1 THEN 6820
6720 IF U6 = 3 OR U6 = 4 THEN 6740: REM 7
6730 GOTO 6920: REM 2

```

```

6740 IF V4 = 2 AND V3 = 6 THEN 6910
6750 GOTO 6970
6760 IF V5 = 4 OR V5 = 5 THEN 6950: REM 5
6770 IF V3 = 7 THEN 6920
6780 GOTO 6930: REM 3
6790 IF U6 = 6 OR U6 = 7 THEN 6950
6800 IF U4 = 2 THEN 6930
6810 GOTO 6970
6820 IF U8 = 7 THEN 6910
6830 GOTO 6930
6840 IF V1 = 3 OR V1 = 6 THEN 6940: REM 4
6850 GOTO 6920
6860 IF V3 = 6 THEN 6990
6870 GOTO 6950
6880 IF U6 = 3 OR U6 = 4 THEN 6970
6890 GOTO 6910: REM 1
6900 PRINT "Numeral is 0":CNX = 0: BOTO 7000
6910 PRINT "Numeral is 1":CNX = 1: BOTO 7000
6920 PRINT "Numeral is 2":CNX = 2: BOTO 7000
6930 PRINT "Numeral is 3":CNX = 3: BOTO 7000
6940 PRINT "Numeral is 4":CNX = 4: BOTO 7000
6950 PRINT "Numeral is 5":CNX = 5: BOTO 7000
6960 PRINT "Numeral is 6":CNX = 6: BOTO 7000
6970 PRINT "Numeral is 7":CNX = 7: BOTO 7000
6980 PRINT "Numeral is 8":CNX = 8: BOTO 7000
6990 PRINT "Numeral is 9":CNX = 9
7000 REM ! CLEAR CHAIN
7010 REM ! CLEAR COMMON
7020 PRINT CHR# (4)"BRUN MAIN.1.OBJ"
23300 REM SPLIT LEFT SIDE
23310 REM UPDATE SEBMENT(split condition)
23320 REM INPUT-II,NP,SP,ML
23330 REM UPDATE PARAMETER
23340 ML = ML + 1: PRINT "Split Segment:";II
23350 FOR U = ML TO II + 2 STEP - 1
23360 LPX(U) = LPX(U - 1):ULX(U) = ULX(U - 1)
23370 E(U) = E(U - 1):DL(U) = DL(U - 1):FL(U) = FL(U - 1)
23380 NEXT U
23390 REM SPLIT IN MIDPOINT OF SEBMENTS
23400 NP = ULX(II):SP = LPX(II)
23410 A1 = INT (NP / 2):A2 = NP - A1: PRINT "A1 = ";A1,"A2 = ";A2
23420 ULX(II) = A1:LPX(II + 1) = SP + A1:ULX(II + 1) = A2
23430 REM RECAL. SEBMENT II,II+1
23440 SP = LPX(II):NP = ULX(II):EP = SP + NP - 1
23450 SOBUB 25000: REM CALL E1
23460 E(II) = E2:DL(II) = DD:FL(II) = FE
23470 SP = LPX(II + 1):NP = ULX(II + 1):EP = SP + NP - 1
23480 SOBUB 25000: REM CALL E1
23490 E(II + 1) = E2:DL(II + 1) = DD:FL(II + 1) = FE
23700 RETURN
24000 REM SPLIT RIGHT SIDE
24010 REM UPDATE SEBMENT(split condition)
24020 REM INPUT-II,NP,SP,MR
24030 REM UPDATE PARAMETER
24040 MR = MR + 1: PRINT "Split Segment:";II
24050 FOR U = MR TO II + 2 STEP - 1
24060 RPX(U) = RPX(U - 1):URX(U) = URX(U - 1)
24070 E(U) = E(U - 1):DR(U) = DR(U - 1):FR(U) = FR(U - 1)
24080 NEXT U
24090 REM SPLIT IN MIDPOINT OF SEBMENTS
24100 NP = URX(II):SP = RPX(II)
24110 A1 = INT (NP / 2):A2 = NP - A1: PRINT "A1 = ";A1,"A2 = ";A2
24120 URX(II) = A1:RPX(II + 1) = SP + A1:URX(II + 1) = A2
24130 REM RECAL. SEBMENT II,II+1
24140 SP = RPX(II):NP = URX(II):EP = SP + NP - 1
24150 SOBUB 26000: REM CALL E1
24160 E(II) = E2:DR(II) = DD:FR(II) = FE
24170 SP = RPX(II + 1):NP = URX(II + 1):EP = SP + NP - 1
24180 SOBUB 26000: REM CALL E1
24210 E(II + 1) = E2:DR(II + 1) = DD:FR(II + 1) = FE
24220 RETURN
25000 REM CALCULATED ERROR NORM,LEFT SIDE
25010 REM BY Integral square error (E2)
25020 REM CAL. n POINT FROM ST TO EP
25030 REM SET PARAMETER
25040 REM X-Vx , Y-Vy , X-Vxx , YY-Vyy , XY-Vxy , FE-angle
25050 REM Vx and Vy
25060 SX = 0:SY = 0:E2 = 0
25070 FOR IP = SP TO EP
25080 SX = SX + LIX(IP):SY = SY + 120 - LJX(IP)
25090 NEXT IP
25100 X = SX / NP:Y = SY / NP
25110 REM Vxx and Vyy
25120 SX = 0:SY = 0
25130 FOR IP = SP TO EP
25140 SX = SX + LIX(IP) * LIX(IP):SY = SY + (120 - LJX(IP)) * (120 - LJX(IP))
25150 NEXT IP
25160 XX = SX - (NP * X * X):YY = SY - (NP * Y * Y)
25170 REM Vxy
25180 SB = 0
25190 FOR IP = SP TO EP
25200 SB = SB + LIX(IP) * (120 - LJX(IP))
25210 NEXT IP
25220 XY = SB - (NP * X * Y)
25230 REM test error=0
25240 IF XY < 0 THEN 25270: REM TEST 45,135
25250 IF (YY - XX) > 0 THEN FE = PI / 2: BOTO 25360
25260 FE = 0: BOTO 25360
25270 IF XX < 0 THEN VY THEN 25300: REM TEST REBULAR COND.
25280 IF XY < 0 THEN FE = PI / 4: BOTO 25360
25290 FE = 3 * PI / 4: BOTO 25360
25300 AG = 0.5 * ATN (2 * XY / (YY - XX))
25310 IF XY > 0 THEN 25340: REM 90<2a<180
25320 IF AG > 0 THEN FE = AG: BOTO 25360
25330 FE = (PI / 2) - ABB (AG): BOTO 25360
25340 IF AG > 0 THEN FE = (PI / 2) + AG: BOTO 25360
25350 FE = PI - ABB (AG)
25360 DD = SIN (FE) * X + COS (FE) * Y:ZZ = FE + 180 / PI
25370 E2 = X * Y + YY - BGR ((X - VY) * (X - VY) + (X - VY) * (Y - VY) + (Y - VY) * (Y - VY)):E2 = ABB (E2 / 2)
25390 RETURN

```

```

24000 REM CALCULATED ERROR NORM,RIGHT SIDE
24010 REM BY integral square error (E2)
24020 REM CAL. n PDINT FROM ST TO EP
24030 REM SET PARAMETER
24040 REM X-Vx , Y-Vy , XX-Vxx , YY-Vyy , XY-Vxy , FE-angle
24050 REM Vx and Vy
24060 SX = 0:SY = 0:E2 = 0
24070 FOR IP = SP TO EP
24080 SX = SX + RIX(IP):SY = SY + 120 - RJX(IP)
24090 NEXT IP
24100 X = SX / NP:Y = SY / NP
24110 REM Vxx and Vyy
24120 SX = 0:SY = 0
24130 FOR IP = SP TO EP
24140 SX = SX + RIX(IP) * RIX(IP):SY = SY + (120 - RJX(IP)) * (120 - RJ
      X(IP))
24150 NEXT IP
24160 KX = SX - (NP * X * X):YY = SY - (NP * Y * Y)
24170 REM Vxy
24180 SB = 0
24190 FOR IP = SP TO EP
24200 SB = SB + RIX(IP) * (120 - RJX(IP))
24210 NEXT IP
24220 XY = SB - (NP * X * Y)
24230 REM test error=0
24240 IF XY < > 0 THEN 26270: REM TEST 45,135
24250 IF (YY - XX) > 0 THEN FE = PI / 2: GOTO 26360
24260 FE = 0: BOTO 26360
24270 IF KX < > YY THEN 26300: REM TEST REGULAR CMD.
24280 IF XY < 0 THEN FE = PI / 4: GOTO 26360
24290 FE = 3 * PI / 4: BOTO 26360
24300 AB = 0.3 * ATN (2 * XY / (YY - XX))
24310 IF XY > 0 THEN 26340: REM 90<Z<180
24320 IF AB > 0 THEN FE = AB: BOTO 26360
24330 FE = (PI / 2) - AB (AB): BOTO 26360
24340 IF AB > 0 THEN FE = (PI / 2) + AB: BOTO 26360
24350 FE = PI - AB (AB)
24360 DD = SIN (FE) * X + COS (FE) * Y:ZZ = FE * 180 / PI
24370 E2 = KX + YY - SQRT ((XX - YY) * (XX - YY) + (4 * XY * XY)):E2 = ABS
      (E2 / 2)
26390 RETURN
27000 REM cal. angle break point
27010 IF DX < > 0 AND DY < > 0 THEN 27060
27020 IF DX = 0 AND DY > 0 THEN AB = PI / 2: BOTO 27130
27030 IF DX = 0 AND DY < 0 THEN AB = 3 * PI / 2: BOTO 27130
27040 IF DY = 0 AND DX > 0 THEN AB = PI: BOTO 27130
27050 IF DY = 0 AND DX < 0 THEN AB = 2 * PI: BOTO 27130
27060 AB = ATN (DY / DX)
27070 IF DY > 0 THEN 27100: REM angle >180
27080 IF AB > 0 THEN AB = 2 * PI - AB: BOTO 27130
27090 AB = PI + ABS (AB): BOTO 27130
27100 REM angle >180
27110 IF AB > 0 THEN AB = PI - AB: BOTO 27130
27120 AB = ABS (AB)
27130 RETURN
35000 REM table zone
35010 IF VV < Y / 5 THEN ZN = 7: BOTO 35060
35020 IF VV < 2 * Y / 5 THEN ZN = 6: BOTO 35060
35030 IF VV < 3 * Y / 5 THEN ZN = 5: BOTO 35060
35040 IF VV < 4 * Y / 5 THEN ZN = 4: BOTO 35060
35050 IF VV < Y THEN ZN = 3: BOTO 35060
35060 RETURN
36000 REM bisection code(0-7)
36010 IF AB < 22.5 OR AB > 337.5 THEN BC = 4: BOTO 36090: REM out
36020 IF AB > 22.5 AND AB < 67.5 THEN BC = 3: BOTO 36090
36030 IF AB > 67.5 AND AB < 112.5 THEN BC = 2: BOTO 36090
36040 IF AB > 112.5 AND AB < 157.5 THEN BC = 1: BOTO 36090
36050 IF AB > 157.5 AND AB < 202.5 THEN BC = 0: BOTO 36090
36060 IF AB > 202.5 AND AB < 247.5 THEN BC = 7: BOTO 36090
36070 IF AB > 247.5 AND AB < 292.5 THEN BC = 6: BOTO 36090
36080 IF AB > 292.5 AND AB < 337.5 THEN BC = 5
36090 RETURN

```

## ประวัติผู้เขียน

นาย พิเชฐ อุดมประเสริฐ เกิดเมื่อวันที่ 20 พฤศจิกายน พ.ศ. 2504  
ที่ อ. เมือง จ. เชียงใหม่ จบการศึกษาระดับปริญญาตรี สาขาวิชา  
วิศวกรรมไฟฟ้า จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ เมื่อปี พ. ศ. 2526

