บรรณานุกรม

1. CALCOMP GROUP, PROGRAMMING CALCOMP ELECTROMACHANICAL PLOTTER, SANDER ASSOCIATES, INC., 1984
2. PREEMAN, R. S., SOFTWARE ENGINEERING A PRACTITION APPROACH, MCGRAW-HILL, INC., 1982
3. FOLEY, J. D., and A. VANDAM, FUNDAMENTALS OF INTERACTIVE COMPUTER GRAPHICS, ADDISON-WESLEY PUBLISHING COMPANY, INC., 1982
4. ROGERS, PROCEDURAL ELEMENTS FOR COMPUTER GRAPHICS, MCGRAW-HILL,INC., 1984.
5. HARRINGTON, S., COMPUTER GRAPHICS A PROGRAMMING APPROACH, MCGRALL-HILL, INC., 1983.
6. RYAN, D. L.,MODERN GRAPHIC COMMUNICATIONS A CAD APPROACH, PRENTICE-HALL INTERNATIONAL EDITIONS, 1986
7. GOODMAN, S. E., and S. T. HEDETNIEMI, INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHMS, p. viii McGRAW-HILL INTERNATIONAL EDITIONS, 1987
8. PRINTRONIX, P-SERIES USER'S REFERENCE MANUAL, PRINTRONIX, INC.,1984
9. HOWE, K., PT200 GRAPHIC OPTION PROGRAMMER'S REFERENCE GUIDE, PRIME COMPUTER, ING., 1985
10. CALAPAI, S., PT200 PROGRAMMER'S REFERENCE GUIDE, PRIME COMPUTER, INC., 1984

ภาคผนวก ก.

ตัวอย่างโปรแกรมลาธิตการเรียกใช้โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม

ตัวอย่างโปรแกรมที่ 1 DEMO01

เป็นการนำเอาโปรแกรมที่ถูกสร้างขึ้นเพื่อแสดงผลลัพธ์ทางพลอตเตอร์มาแสดงผลลัพธ์ทางเครื่องพิมพ์แทน โดยนำโปรแกรมที่มีอยู่มาคอมไพล์ (COMPILE) และ เชื่อมต่อ (LINK) กับโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรมนี้แทน (โปรดดูในตัวอย่างผลลัพธ์ที่ 1)

ตัวอย่างโปรแกรมที่ 2 DEMO02

เป็นการสาธิต วิธีเรียกใช้โปรแกรมย่อยทางกราฟฟิก สำหรับสร้างรูปกราฟเส้นและกราฟจุด ใช้ในกรณีที่ข้อมูลมีจำนวนน้อย ต่อไปนี้จะเป็นการอธิบายตัวโปรแกรม

```
      CALL PLOTS (0,0,0)
      ( ใช้เริ่มต้นการเรียกใช้โปรแกรมย่อยทางกราฟฟิก )
      REAL* 4 DATAX(11), DATAY (11)
```

(กำหนดข้อมูลที่จะทำการพลอต ในที่นี้จะมี 9 ข้อมูล แต่ที่ต้องกำหนดไว้ 11 ข้อมูลก็เพื่อไว้ให้สำหรับโปรแกรมย่อย SCALE ไว้สำหรับเก็บข้อมูลที่บอกค่าที่น้อยที่สุดในชุดข้อมูล และอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาว)

```
      DO 100  I = 1, NDATA
100   DATAX(I) = REAL(I)
      ( กำหนดข้อมูลแทนแนวราบมีค่าตั้งแต่ 1 ถึง 9 )
      DATAY (1)      = 5.
      DATAY (2)      = 1.
      DATAY (3)      = 4.8
      DATAY (4)      = 2.7
      DATAY (5)      = 4.
      DATAY (6)      = 0.9
```

```
DATAY (7)       = 2.4
DATAY (8)       = 1.7
DATAY (9)       = 3.6
```

(กำหนดข้อมูลแกนแนวดิ่ง (แนว-Y))

```
NDATA = 9
```

(กำหนดจำนวนข้อมูลที่จะทำการพลอต)

```
XAXLENG         = 8.
YAXLENG         = 4.
```

(กำหนดความยาวแกนแนวราบ (แนว-X) และแกนแนวดิ่ง (แนว-Y))

```
INC             = 1
```

(กำหนดระยะห่างข้อมูลที่เก็บในชุดข้อมูล ในที่นี้ข้อมูลจะเก็บเรียงต่อกันไปจนครบ จึงกำหนดให้มีค่าเท่ากับ 1)

```
CALL SCALE (DATAX, XAXLENG, NDATA, INC)
CALL SCALE (DATAY, YAXLENG, NDATA, INC)
```

(ทำการคำนวณหาค่าที่น้อยที่สุดในชุดข้อมูล(ARRAY)จะเก็บไว้ที่ตำแหน่ง NDATA*INC+1 และ อัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาวจะเก็บไว้ที่ตำแหน่ง NDATA*INC+INC+1 โดยคำนวณในแต่ละชุดข้อมูลของแต่ละแกนแยกกัน)

```
CALL AXIS (1., 1., 'X-AXIS',-6, XAXLENG., 0.,
*      DATAX(NDATA*INC+1), DATAX(NDATA*INC+INC+1) )
```

(โปรแกรมย่อยนี้ ทำการตั้งแกนเริ่มต้นไว้ที่ตำแหน่ง (1., 1.) มีคำอธิบายว่า 'X-AXIS' ซึ่งมีจำนวน 6 ตัวอักษร แต่ที่ใช้ค่า -6 หมายความว่า คำอธิบายนี้ต้องการพลอตไว้ใต้แกนกราฟ XAXLENG กำหนดความยาวแกนกราฟ ค่า 0. ใช้กำหนดมุมของแกนกราฟในที่นี้เป็นแกนแนวราบจึงกำหนดค่าไว้เท่ากับ 0. ตัวแปร DATAX(NDATA*INC+1) กำหนดค่าน้อยที่สุดในชุดข้อมูลซึ่งจะถูกกำกับแกนกราฟในตอนต้นแกน ส่วนตัวแปร DATAX(NDATA*INC+INC+1) กำหนดอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาว จะถูกนำมาใช้คำนวณค่ากำกับแกนกราฟ

เป็นการบอกตำแหน่งเวลาพลอตกราฟว่ามีค่าจริงของข้อมูลคือเท่าไร)

```
CALL AXIS (1., 1., 'Y-AXIS', 6, YAXLENG, 90.,
DATAY (NDATA*INC+1), DATAY (NDATA*INC+INC+1) )
```

(โปรแกรมย่อยนี้ทำการตั้งแกนเริ่มต้นไว้ที่ ตำแหน่ง (1., 1.) มีคำอธิบายว่า 'Y-AXIS' ซึ่งมีจำนวน 6 ตัวอักษร ใช้ค่า -6 เป็นบวกหมายความว่าคำอธิบายนี้ต้องการพลอตไว้เหนือแกนกราฟ YAXLENG กำหนดความยาวของแกนกราฟ ค่า 90. ใช้กำหนดมุมของแกนกราฟ ในที่นี้เป็นแกนแนวดิ่งจึงกำหนดว่าไว้เท่ากับ 90 องศา ตัวแปร DATAY(NDATA*INC+1) กำหนดค่าน้อยที่สุดในชุดข้อมูลซึ่งจะถูกนำมาอธิบายกำกับแกนกราฟในตอนต้นแกน ส่วนตัวแปร DATAY (NDATA*INC+INC+1) กำหนดอัตราส่วนของค่าของข้อมูลต่อหนึ่งหน่วยความยาวจะถูกนำมาใช้คำนวณค่ากำกับแกนกราฟ เป็นการบอกตำแหน่งเวลาพลอตกราฟว่ามีค่าจริงของข้อมูลคือเท่าไร)

```
CALL PLOT (1., 1., -3)
```

(เป็นการย้ายจุดกำเนิด (XORG, YORG) เพราะโปรแกรมย่อย LINE จะอ้างอิงกับจุดกำเนิดเสมอในการพลอตเส้น จึงย้ายจุดกำเนิดมาไว้ที่จุดตัดของแกนกราฟ (1., 1.))

```
CALL LINE (DATAX,DATAY,NDATA,INC,LINETYPE,CHARTYPE)
LINETYPE = 5
```

(กำหนดลักษณะของเส้น (โปรดดูในวิธีใช้โปรแกรมย่อย DASHLN))

```
CHARTYPE = 5
```

(กำหนดลักษณะของตัวอักษรพิเศษที่ใช้กำกับแต่ละจุดข้อมูล มีตั้งแต่ 0 ถึง 9 (โปรดดูในตัวอย่างผลลัพธ์ที่ 3))

(โปรแกรมย่อย LINE นี้จะทำการพลอตเส้นตามข้อมูลที่เข้ามาโดยคำนวณตำแหน่งใหม่ให้เหมาะสมกับแกนกราฟ คือนำข้อมูลที่จะพลอตมาลบกับค่าที่น้อยที่สุดก่อน แล้วคูณด้วยอัตราส่วนค่าของข้อมูลต่อหน่วยความยาวจึงเป็นตำแหน่งที่ทำการพลอต หลังจากนี้จะได้ผลลัพธ์เป็นรูปกราฟเส้นเรียบร้อย)

```
DO 1000 A=0., 0.03, 0.0125
CALL PLOT (0., 0.03, 0.0125
CALL SYMBOL (3.+A,5.,5, `LINE CHART', IDUMMY, 0.,10)
(ทำการการพลอตคำอธิบายรูปกราฟทั้งหมดโดยพลอตเป็นตัวอักษรตัวหนา)
CALL PLOT (0., 0., 999)
( สั่งเลิกการเรียกใช้ โปรแกรมย่อยทางกราฟฟิก )
STOP
END
( จบการทำงานของโปรแกรม )
```

ตัวอย่างโปรแกรมที่ 3 DEM003

เป็นโปรแกรมที่จะแสดงตัวอักษรที่โปรแกรมย่อยทางกราฟฟิกมีอยู่เป็นตัวอักษร ASCII และตัวอักษรพิเศษที่ใช้ในการพลอตกราฟ (โปรดดูตัวอย่างผลลัพธ์ที่ 3 ประกอบ)

ตัวอย่างโปรแกรมที่ 4 DEM004

เป็นโปรแกรม สาธิตการสร้างรูปหลายเหลี่ยม (POLYGON) และมีเส้นตรงเชื่อมระหว่างจุดแต่ละจุดด้วย

```
WRITE (1,*) 'HOW MANY POINTS? :'
READ  (1,*)  NPTS
```

(ใช้ถามจำนวนจุดมุมของรูปหลายเหลี่ยม เช่นถ้ากำหนดมาเท่ากับ 3 ก็จะเป็นรูป สามเหลี่ยม ฯลฯ จำนวนจุดนี้ต้องไม่เกิน 40 จุด เพราะเตรียมที่ไว้ 40 จุด ตามที่กำหนดข้างต้น)

```
REAL * 4 X(40), Y(40)
DO 20 I= 1, NPTS
```

```
      X(I) = 5.25 + 3.75*SIN (I*6.2832 NPTS)
```

(ค่า 5.25 คือ ตำแหน่งแนวราบของจุดศูนย์กลางของวงกลมที่รูปหลายเหลี่ยมนี้จะอยู่ภายใน ค่า 3.75 คือ รัศมีของวงกลม ค่า 6.2832 คือ ค่า 2 * PIE หรือมุมรอบจุดศูนย์กลางในหน่วยเรเดียน)

```
20    Y(I) = 3.00+3.75*COS (I*6.2832/NPTS)
```

(ค่า 3.00 คือ ตำแหน่งแนวดิ่งของจุดศูนย์กลางของวงกลมที่รูปหลายเหลี่ยมจะอยู่ภายใน ทั้งสองบรรทัดนี้จะทำการคำนวณหาตำแหน่งของแต่ละจุดของรูปหลายเหลี่ยมที่จะอยู่บนเส้นรอบวงของวงกลม)

```
      CALL PLOTS (0, 0, 0)
```

(เริ่มการเรียกใช้โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม)

```
      DO 40 I = 1,NPTS-1
```

(เพราะจำนวนเส้นที่จะมีการลากเส้นเชื่อมจุดแต่ละจุดทั้งหมด จะเท่ากับ NPTS-1 เนื่องเส้นที่ลากจากจุดตัวเองมายังจุดตัวเองไม่ต้องมี)

```
      K = I+1
      DO 30  J = K,NPTS
      CALL PLOT (X(I), Y(I), 3)
      CALL PLOT (X(J), Y(J), 2)
30 CONTINUE
40 CONTINUE
```

(ทำการพลอตทุกเส้นที่เชื่อมทุกๆจุดเข้าด้วยกัน)

```
      CALL PLOT (0., 0., 999)
```

(เลิกการใช้โปรแกรมย่อยทางกราฟฟิก)

```
      STOP
      END
```

(จบการทำงานของโปรแกรม)

โปรดดูตัวอย่างผลลัพธ์ที่ 4 แสดงผลลัพธ์ของรูป 10 เหลี่ยม

ตัวอย่างโปรแกรมที่ 5 DEM005

เป็นโปรแกรมสร้างกราฟแท่งเปรียบเทียบของข้อมูล 8 คู่ข้อมูลเทียบกันต่อไปนี้จะเป็นตัวโปรแกรม

```
CALL PLOTS (0, 0, 0)
( เริ่มต้นการใช้โปรแกรมย่อยทางกราฟฟิก )
REAL*4 DATA1 (8), DATA2 (8)
DATA1 (1)       = 2.
DATA1 (2)       = 1.
DATA1 (3)       = 5.
DATA1 (4)       = 3.
DATA1 (5)       = 4.
DATA1 (6)       = 0.3
DATA1 (7)       = 2.6
DATA1 (8)       = 4.9
DATA2 (1)       = 1.7
DATA2 (2)       = 3.4
DATA2 (3)       = 2.6
DATA2 (4)       = 2.9
DATA2 (5)       = 1.7
DATA2(6)        = 0.4
DATA2 (7)       = 3.3
DATA2 (8)       = 4.6
( กำหนดค่าให้ข้อมูลในชุดข้อมูล )
XAXLENG         = 8.
YAXLENG         = 5.
( กำหนดความยาวของแกนกราฟทั้งสอง )
```

```
      NDATA          = 8
      ( กำหนดจำนวนคู่ข้อมูลที่จะทำากรพลอตกราฟแท่งเปรียบเทียบ )
      CALL AXIS (1., 1., 'X-AXIS', -6, XAXLENG, 0., 0., 1.)
      CALL AXTS (1., 1., 'Y-AXIS', 6, YAXLENG, 90., 0., 1.X
      ( สร้างแกนกราฟทั้งแนวดิ่งและแนวราบใช้อ้างอิงในการอ่านค่าข้อมูลจาก
กราฟแท่งเปรียบเทียบ )
      CALL BAR2 (1., 1., DATA1, DATA2, NDATA, XAXLENG)
      ( สร้างกราฟแท่งเปรียบเทียบ ค่า 1., 1. คือตำแหน่งจุดตัดของแกน
กราฟแนวราบและแนวดิ่ง ตามโปรแกรมย่อย AXIS ข้างบน )
      DO 1000 A= 0., 0.08, .0125
1000  CALL SYMBOL (3.+A,5.,.5, 'BAR CHART',IDUMMY,0.,9)
      ( ทำการพลอตคำอธิบายรูปกราฟทั้งหมดโดยพลอตเป็นตัวอักษรหนา )
      CALL PLOT (0., 0., 999)
      ( สั่งเลิกใช้โปรแกรมย่อยทางกราฟฟิก )
      (โปรดดูในตัวอย่างผลลัพธ์ที่ 5)
```

ตัวอย่างโปรแกรมที่ 6 DEM006

เป็นโปรแกรมลาธิตการพลอตกราฟวงกลม ต่อไปนี้จะเป็นการอธิบายตัวโปรแกรม

```
      CALL PLOTS (0, 0, 0)
      ( ใช้เริ่มต้นการเรียกใช้โปรแกรมย่อยทางกราฟฟิก )
      REAL*4 DATA (10)
      ( กำหนดจำนวนข้อมูลที่จะทำกราฟรูปวงกลม )
      DATA (1)       = 2.
      DATA (2)       = 1.
      DATA (3)       = 5.
      DATA (4)       = 3.
```

```
      DATA (5)        = 4.
      DATA (6)        = 2.5
      DATA (7)        = 4.3
      DATA (8)        = 3.4
      DATA (9)        = 4.5
      DATA (10)       = 1.2
      ( กำหนดค่าใช้กับข้อมูลในชุดข้อมูล )
      XC              = 5.
      YC              = 3.5
      ( กำหนดจุดศูนย์กลางของรูปกราฟวงกลม )
      R               = 2
      ( กำหนดรัศมีของรูปวงกลม )
      IMODE           = 1
      ( กำหนดโหมดรูปกราฟวงกลมที่ไม่แยกส่วนของวงกลมออกมา )
      CALL PIE (XC, YC, DATA, NDATA, R, IMODE)
      ( พลอตรูปกราฟวงกลมตามต้องการ )
      CALL CSLANT (4.5)
      ( กำหนดมุมเอียงของตัวอักษรที่ใช้อธิบายรูปกราฟวงกลม )
      DO 1000 A = 0., 0.08, 0.0125
1000  CALL SYMBOL (3.+A, 5., .5, 'PIE CHART', IDUMMY, 0.,90
      ( ทำการพลอตคำอธิบายรูปกราฟทั้งหมดโดยพลอตเป็นตัวอักษรหนา )
      CALL PLOT (0., 0., 999)
      ( สั่งเลิกใช้โปรแกรมย่อยทางกราฟฟิก )
      (โปรดดูในตัวอย่างผลลัพธ์ที่ 6)
      STOP
      END
```

ตัวอย่างโปรแกรมที่ 7 DEMO07

เป็นโปรแกรมสาธิตการพลอตกราฟวงกลม ต่อไปนี้จะเป็นการอธิบายตัวโปรแกรม

```
CALL PLOTS (0, 0, 0)
( ใช้เริ่มต้นการเรียกใช้โปรแกรมย่อยทางกราฟฟิก )
REAL*4 DATA(10)
( กำหนดจำนวนข้อมูลที่จะแสดงผลลัพธ์ทางกราฟรูปวงกลม )
DATA (1)        = 2.
DATA (2)        = 1.
DATA (3)        = 5.
DATA (4)        = 3.
DATA (5)        = 4.
DATA (6)        = 2.5
DATA (7)        = 4.3
DATA (8)        = 3.4
DATA (9)        = 4.5
DATA (10)       = 1.2
( กำหนดค่าให้กับข้อมูลในชุดข้อมูล )
XC              = 5.
YC              = 3.5
( กำหนดจุดศูนย์กลางของรูปกราฟวงกลม )
R               = 2.
( กำหนดรัศมีของรูปกราฟวงกลม )
IMODE           = 2.
( กำหนดโหมดรูปกราฟวงกลมโดยแยกแต่ละส่วนของวงกลมออกมา )
```

```
      CALL PIE (XC, YC, DATA, NDATA, R, IMODE)
      ( พลอตรูปกราฟวงกลมตามต้องการ )
      CALL CSLANT (4.5)
      ( กำหนดมุมเอียงของตัวอักษรที่ใช้อธิบายรูปกราฟวงกลม )
      DO 1000 A= 0., .08, .0125
1000  CALL SYMBOL (3+A,5.,.5,'SPLIT PIE CHART', IDUMMY, 0,15)
      ( ทำการพลอตคำอธิบายรูปกราฟทั้งหมดโดยพลอตเป็นอักษรหนา )
      CALL PLOT CO>< 0, 999)
      ( สั่งเลิกใช้โปรแกรมย่อยทางกราฟฟิก )
      (โปรดดูในตัวอย่างผลลัพธ์ที่ 7)
      STOP
      END
      ( จบการทำงานของโปรแกรม )
```

ตัวอย่างโปรแกรมที่ 8 DEMO08

เป็นการสาธิตวิธีการเรียกใช้โปรแกรมย่อยทางกราฟฟิกสำหรับสร้างรูปกราฟเส้นจากข้อมูลที่มีปริมาณมากซึ่งเก็บไว้ในแฟ้มข้อมูล ต่อไปนี้จะเป็นการอธิบายตัวโปรแกรม

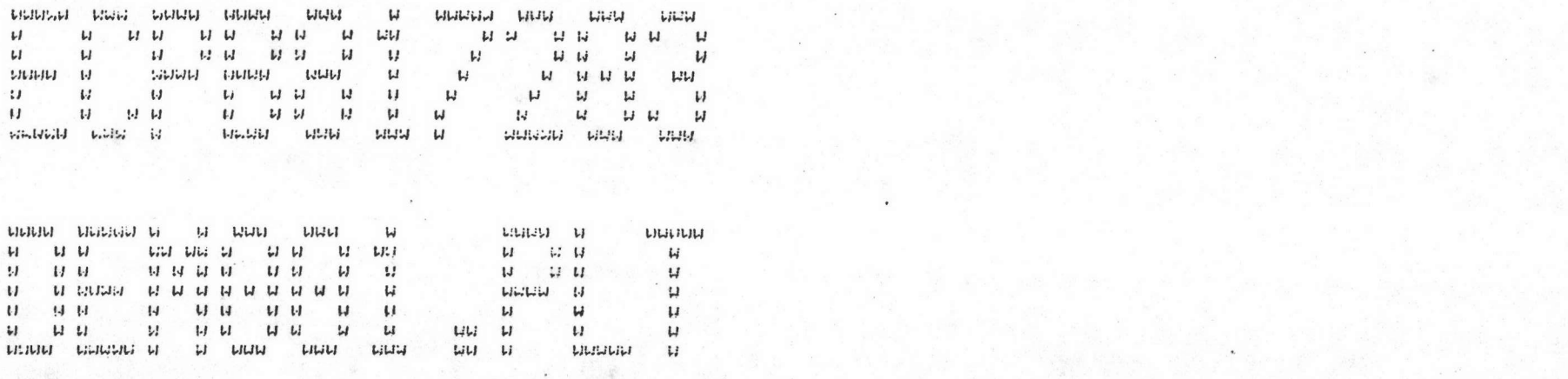
```
      CALL PLOTS (0,0,0)
      ( เริ่มต้นการเรียกใช้โปรแกรมย่อยทางกราฟฟิก )
      OPEN (71, FILE= 'MORSUN.PRN')
      FORMAT (4x, F4.2,3X,F4.2)
      ( เปิดแฟ้มข้อมูลที่เก็บข้อมูลที่ต้องการพลอตกราฟ )
      DO 100 J = 1,NDATA
```

```
100     READ (71,17) DATAX (J), DATA (J)
        ( อ่านข้อมูลเข้ามาเก็บไว้ในชุดข้อมูลที่กำหนดไว้ )
        REAL*4 DATAX (482), DATAY(482)
        ( ที่กำหนด 482 เพราะต้องสำรองอีก 2 ที่สำหรับเก็บค่าน้อยที่สุดของข้อมูล
ในชุดข้อมูลและอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาว )
        NDATA   = 480
        ( กำหนดจำนวนข้อมูลที่จะทำการพลอต )
        INC     = 1
        (  กำหนดระยะห่างข้อมูลที่เก็บในชุดข้อมูลในที่นี้จะเก็บเรียงต่อกัน
        จึงกำหนดให้เท่ากับ 1 )
        XAXLENG = 8.
        YAXLENG = 6.
        ( กำหนดความยาวของแกนกราฟทั้งสองแนว แนวดิ่งและแนวราบ )
        CALL SCALE (DATAX, XAXLENE, NDATA, INC)
        CALL SCALE (DATAY, YAXLENG, NDATA, ING)
        ( คำนวณหาค่าที่น้อยที่สุดของข้อมูลในชุดข้อมูลและอัตราส่วนค่าของข้อมูล
ต่อหนึ่งหน่วยความยาว )
        CALL AXIS (1.,1.,'TIME-AXIS',-9,XAXLENG,
     *      DATAX(NDATA*ING +1),DATAX(NDATA*INC+INC+1) )
        CALL AXIS (1.,1.,'MAN-AXIS',8,YAXLENG,
     *      DATAY(NDATA*INC+1),DATAY(NDATA*INC+INC+1) )
        (พลอตแกนกราฟทั้งแนวราบและแนวดิ่งไว้สำหรับอ้างอิงในการอ่านข้อมูล)
        CALL PLOT ( 1., 1., -3)
        CALL LINE (DATAX, DATAY, NDATA, INC, 0, 0)
        ( พลอตกราฟเส้น )
```

```
      DO 1000 A= 0., 0.08, 0.0125
1000  CALL SYMBOL (3.+A,5.,.5, 'LINE CHART',IDUMMY, 0.,10)
      ( พลอตคำอธิบายกราฟ )
      CALL PLOT (0., 0., 999)
      STOP
      END
      ( จบการทำงานของโปรแกรม )
```

User: ECPR817203 -at PRO

<STUD>THESIS>ECPR817203>F77>CLCNTEST>DEMO01.PLT



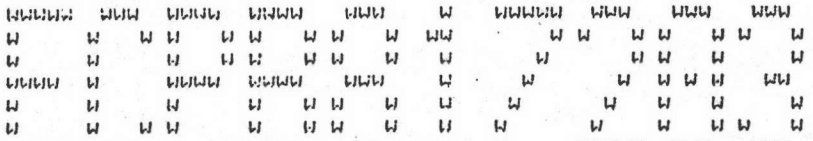
Label: PRTQ20 -form

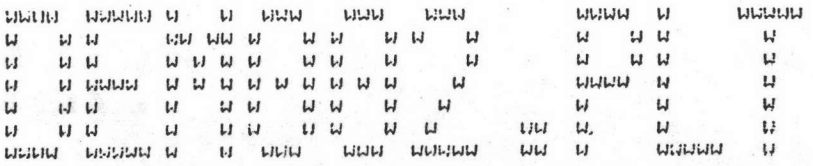
Pathname: <STUD>THESIS>ECPR817203>F77>CLCNTEST>DEMO01.PLT
File last modified: 87-09-01.15:30:52.Tue

Spooled: 87-09-18.10:44:52.Fri [Spooler rev 19.4]
Started: 87-09-18.10:59:24.Fri on: PRO by: PRO

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>DEM002.PLT





Label: PRT021 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>DEM002.PLT
File last modified: 87-09-01.15:34:32.Tue

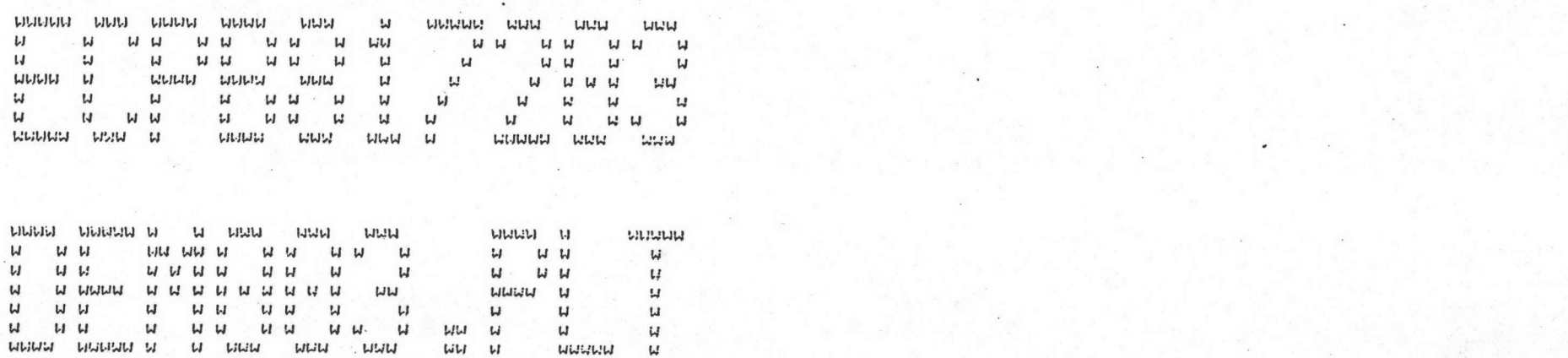
Spooled: 87-09-18.10:45:04.Fri [Spooler rev 19.4]
Started: 87-09-18.11:01:40.Fri on: PRO by: PRO

GRAPHIC OUTPUT

รูปที่ ก-2 แสดงตัวอย่างผลลัพธ์ที่ 2

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>DEM003.PLT



Label: PRT022 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>DEM003.PLT
File last modified: 87-08-31.17:41:56.Mon

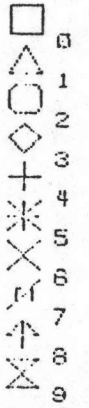
Spooled: 87-09-18.10:45:18.Fri [Spooler rev 19.4]
Started: 87-09-18.11:02:40.Fri on: PRO by: PRO

BASIC Character Set

ASCII Characters

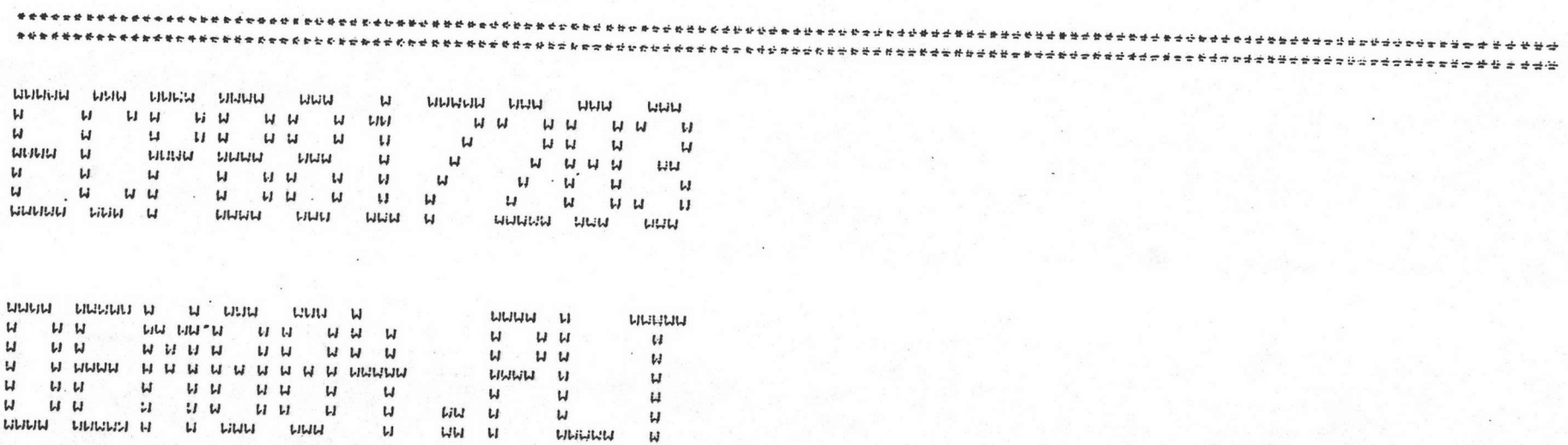
240	256	< 274	J 312	X 330	f 346	† 364
! 241	/ 257	= 275	K 313	Y 331	g 347	u 365
" 242	0 260	> 276	L 314	Z 332	h 350	v 366
# 243	1 261	? 277	M 315	[333	i 351	w 367
$ 244	2 262	@ 300	N 316	\ 334	j 352	x 370
% 245	3 263	A 301	O 317	] 335	k 353	y 371
& 246	4 264	B 302	P 320	↑ 336	l 354	z 372
' 247	5 265	C 303	Q 321	← 337	m 355	
(250	6 266	D 304	R 322	340	n 356	
) 251	7 267	E 305	S 323	a 341	o 357	
* 252	8 270	F 306	T 324	b 342	p 360	
+ 253	9 271	G 307	U 325	c 343	q 361	
, 254	: 272	H 310	V 326	d 344	r 362	
- 255	; 273	I 311	W 327	e 345	s 363	

Centered Symbols



User: ECPB817203 -at PRO

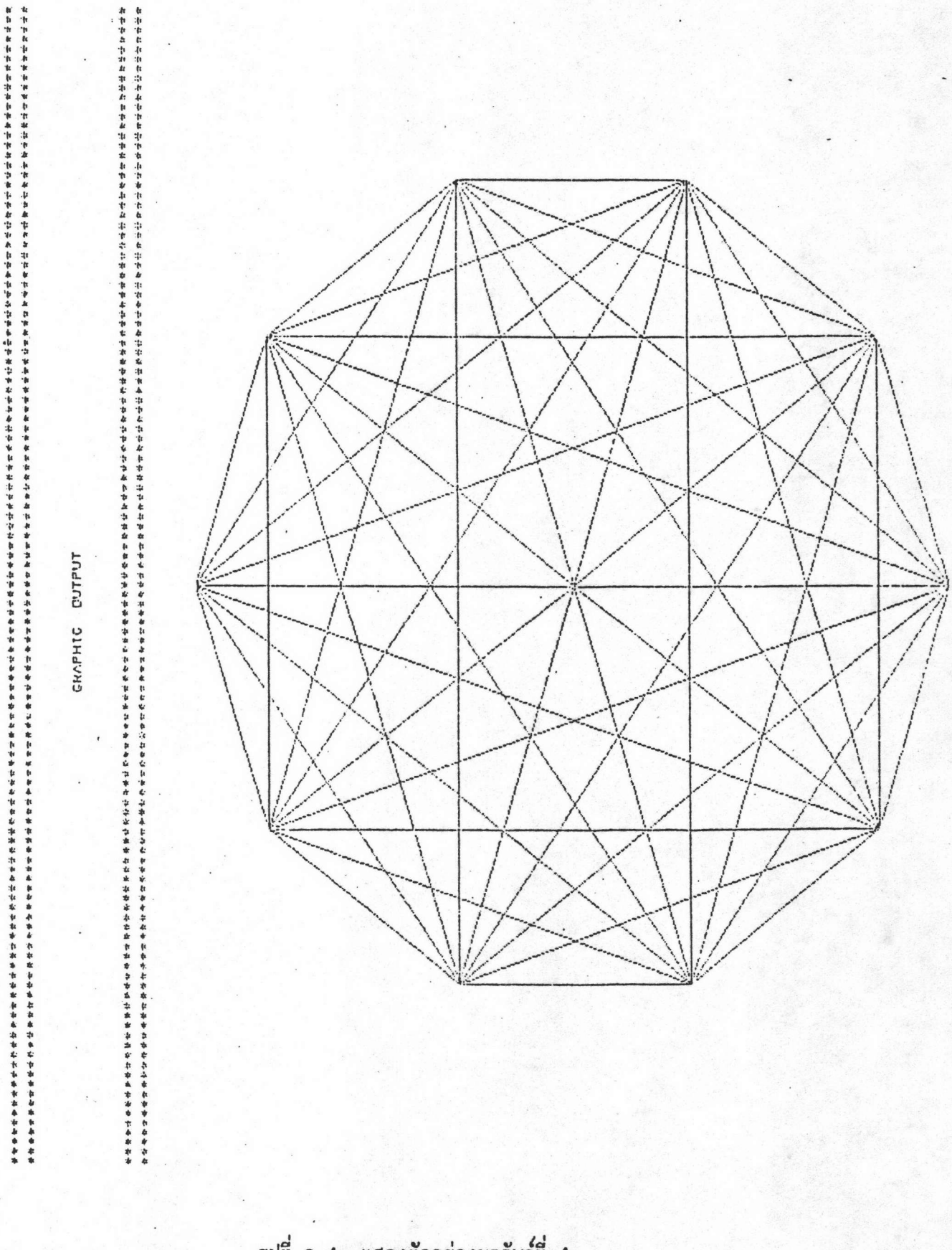
<STUD>THESIS>ECPB817203>F77>CLCMTEST>DEMO04.PLT



Label: PRT023 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>DEMO04.PLT
File last modified: 87-09-01.15:37:52.Tue

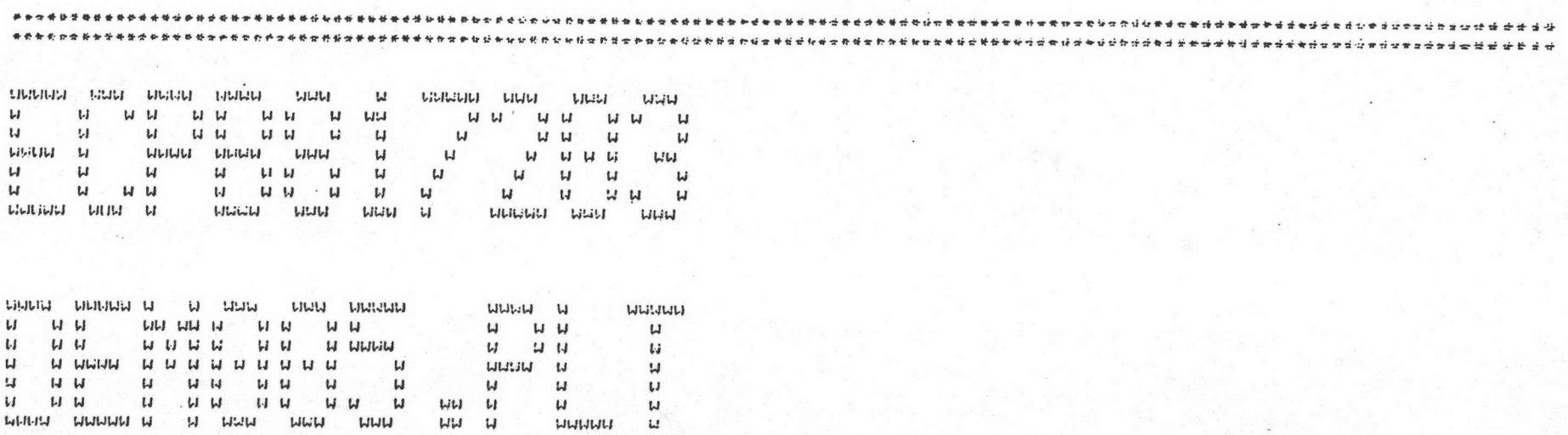
Spooled: 87-09-18.10:45:28.Fri [Spooler rev 19.4]
Started: 87-09-18.11:03:40.Fri on: PRO by: PRO



รูปที่ ก-4 แสดงตัวอย่างผลลัพธ์ที่ 4

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCNTEST>DEMO05.PLT

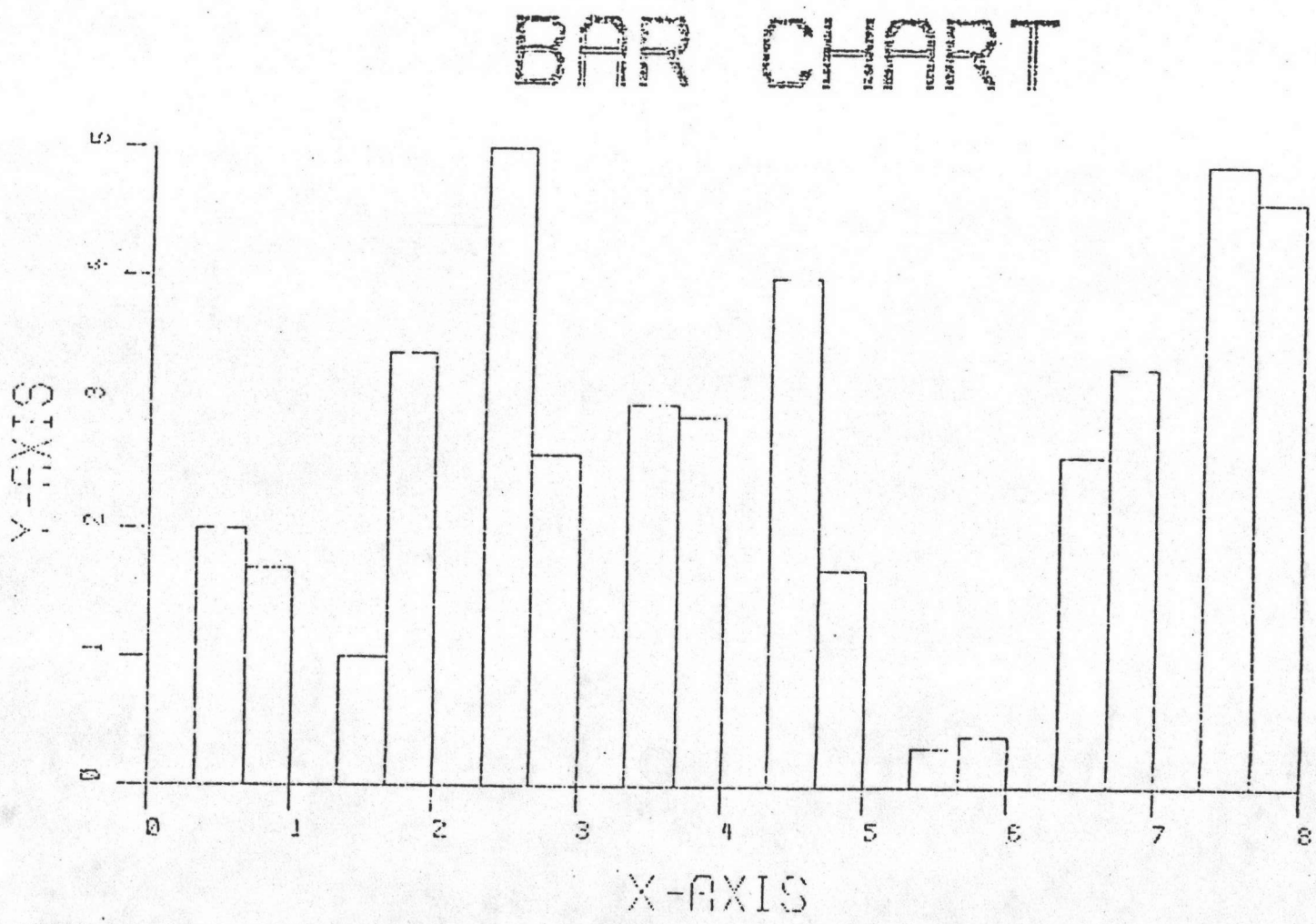


Label: PRT024 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCNTEST>DEMO05.PLT
File last modified: 87-09-01.15:43:12.Tue

Spooled: 87-09-18.10:45:40.Fri [Spooler rev 19.4]
Started: 87-09-18.11:04:40.Fri on: PRO by: PRO

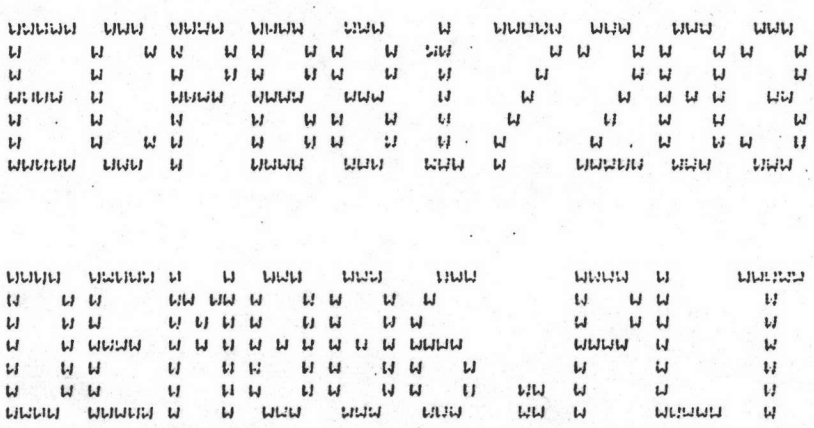
GRAPHIC OUTPUT



รูปที่ ก-5 แสดงตัวอย่างผลลัพธ์ที่ 5

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCNTEST>DEM006.PLT



Label: PRT025 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCNTEST>DEM006.PLT
File last modified: 87-09-01.15:48:08.Tue

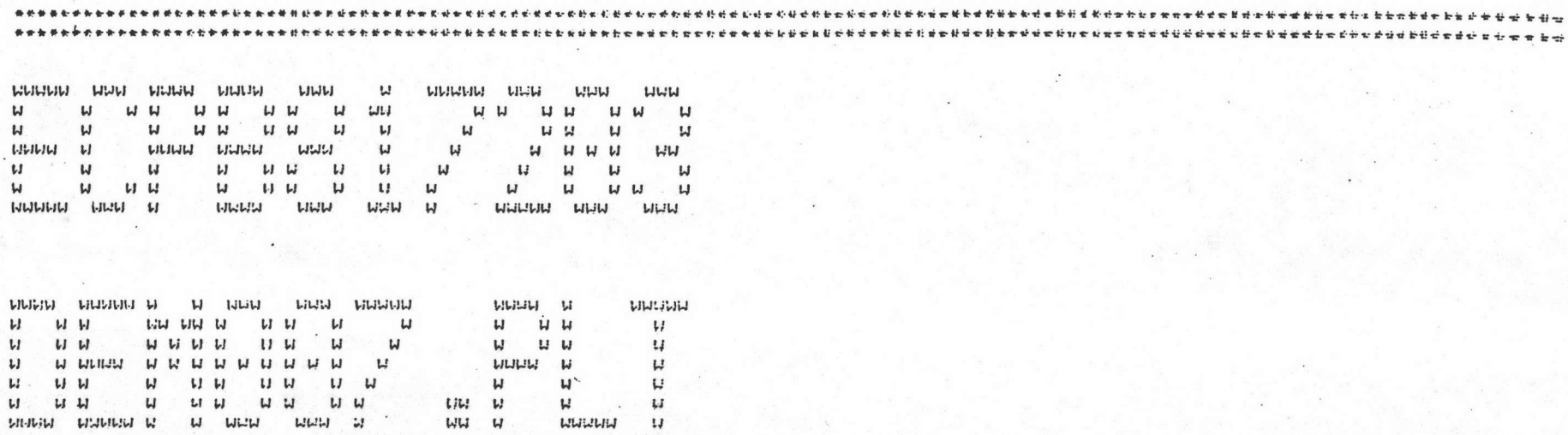
Spooled: 87-09-18.10:45:48.Fri [Spooler rev 19.4]
Started: 87-09-18.11:05:36.Fri on: PRO by: PRO

GRAPHIC OUTPUT

รูปที่ ก-6 แสดงตัวอย่างผลลัพธ์ที่ 6

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>DEM007.PLT

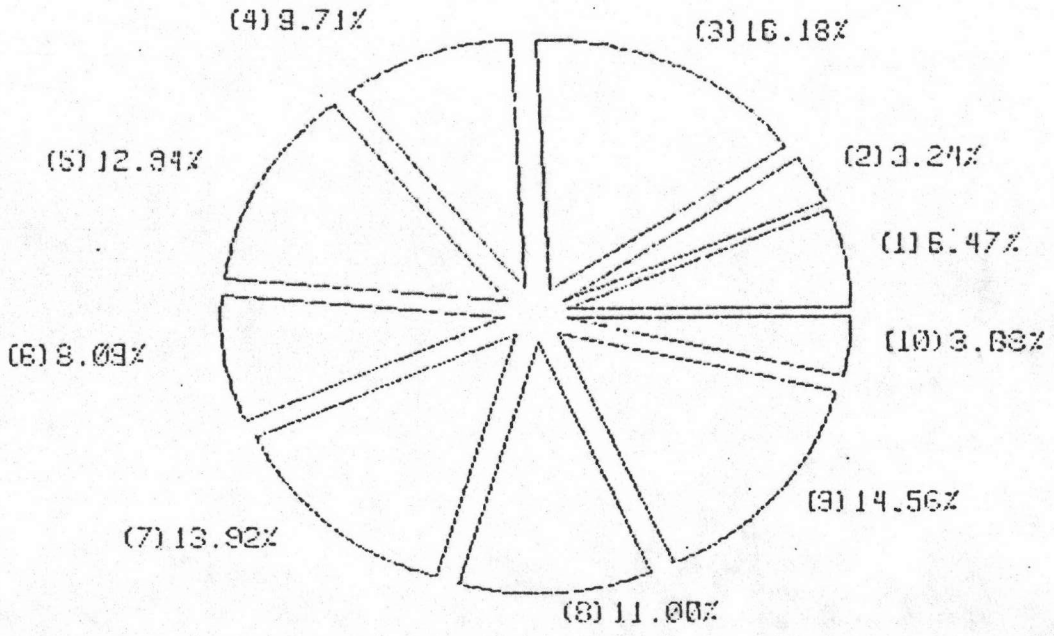


Label: PRT026 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>DEM007.PLT
File last modified: 87-09-01 15:55:20 Tue

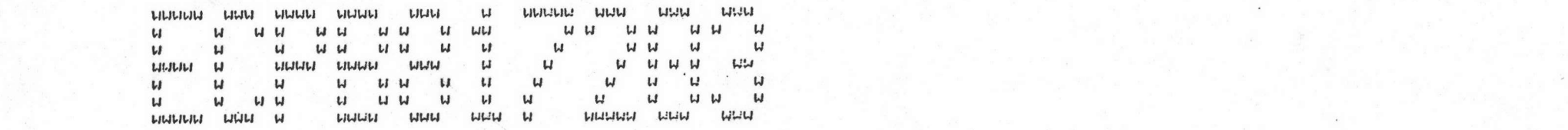
Spooled: 87-09-18 10:45:56 Fri [Spooler rev 19.4]
Started: 87-09-18 11:06:40 Fri on: PRO by: PRO

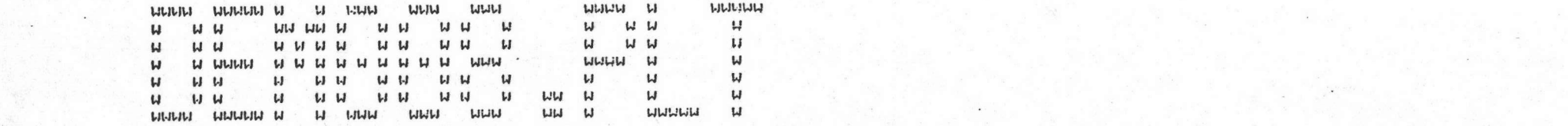
SPLIT PIE CHART



User: ECP0317203 -at PRO

<STUD>THESIS>ECP0317203>F77>CLCNTEST>DEM008.PLT





Label: PRT027 -form

Pathname: <STUD>THESIS>ECP0317203>F77>CLCNTEST>DEM008.PLT
File last modified: 87-09-01.15:59:04.Tue

Spooled: 87-09-18.10:46:12.Fri [Spooler rev 19.4]
Started: 87-09-18.11:07:32.Fri on: PRO by: PRO

รูปที่ ก-8 แสดงตัวอย่างผลลัพธ์ที่ 8

ภาคผนวก ข.

อุปกรณ์แสดงผลลัพธ์ทางกราฟฟิก

ข-1.1 เครื่องพิมพ์[๘]

เครื่องพิมพ์ที่ใช้เป็นผลิตภัณฑ์ที่มีเครื่องหมายการค้าว่า PRINTRONIX เป็นเครื่องพิมพ์แบบ ดอต-เมตริกซ์ (dot matrix) โดยทำการพิมพ์ครั้งละหนึ่งบรรทัด (line printer) โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรมใช้เครื่องพิมพ์นี้ในพลอตโหมด (plot mode) ซึ่งสามารถสร้างรูปภาพได้ตามต้องการ การทำเครื่องพิมพ์ให้อยู่ในโหมดพลอต โดยอาศัยการส่งรหัสควบคุม (รหัสควบคุมจะเท่ากับ 5) ไปยังเครื่องพิมพ์ทุก ๆ ระเบียนข้อมูล (record)

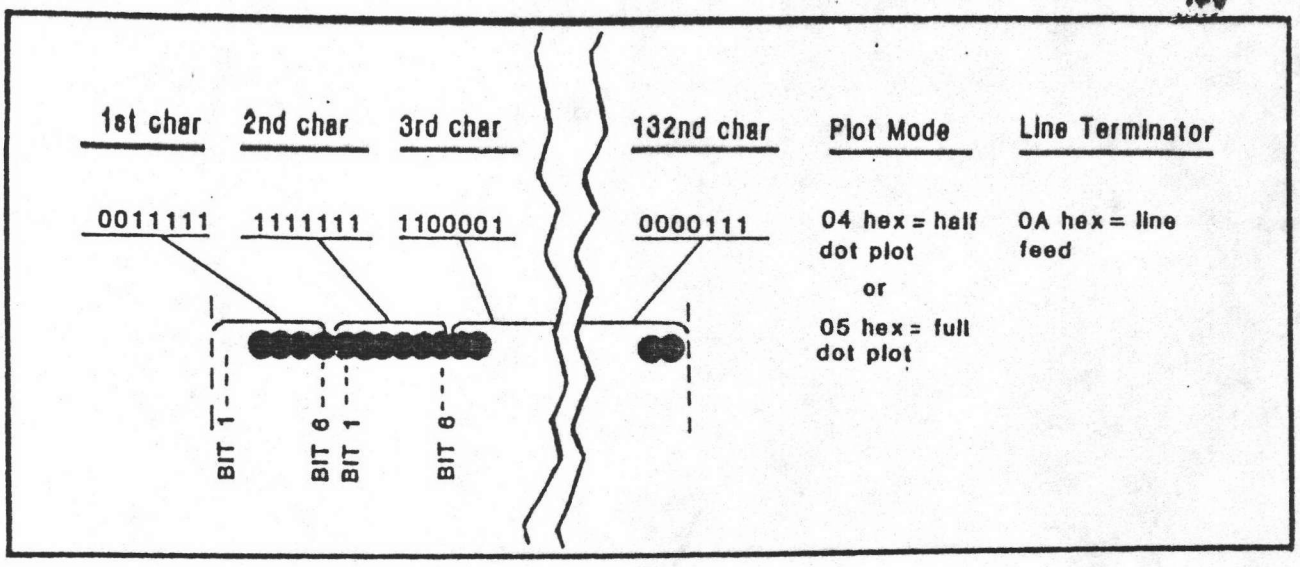
รูปแบบของระเบียนข้อมูลไปที่แรกจะเป็นรหัสควบคุม ไบท์ต่อ ๆ ไปจะเป็นข้อมูลจุดที่ต้องพลอต (bit map) โดยใน 1 ไบท์ จะเก็บข้อมูลของจุดได้ 8 จุดโปรแกรมย่อยทางกราฟฟิก จะใช้ 120 ไบท์ต่อหนึ่งระเบียนและไบท์สุดท้ายจะเป็นรหัสบอกให้ขึ้นบรรทัดใหม่ หรือระบียนต่อไป (line feed) จะมีทั้งหมด 600 ระเบียนที่ใช้ รูปแบบของรหัสข้อมูลควบคุมเครื่องพิมพ์จะเป็นดังรูปที่ ข-1

ข-1.2 การสร้างโปรแกรมควบคุมการแสดงผลลัพธ์ทางเครื่องพิมพ์

เนื่องจาก คำสั่งพลอตที่รับเข้ามายังอุปกรณ์แสดงผลลัพธ์จำลอง ไม่สามารถนำออกแสดงผลลัพธ์ทางเครื่องพิมพ์ได้ทันที เพราะฉะนั้นการแสดงผลลัพธ์จะทำได้ก็ต่อเมื่อ การพลอตรูปภาพเสร็จสิ้นลงแล้ว คำสั่งพลอตจะถูกเก็บลงไว้ในแฟ้มข้อมูลก่อน เมื่อมีความต้องการผลลัพธ์แสดงออกทางเครื่องพิมพ์ จึงจะนำคำสั่งพลอตของอุปกรณ์แสดงผลลัพธ์จำลองที่เก็บไว้ในแฟ้มข้อมูลออกมาทีละคำสั่ง แต่เนื่องจากเครื่องพิมพ์ไม่สามารถเลื่อนกระดาษลงย้อนกลับได้ ทำให้ต้องทำการพลอตลงบนหน่วยความจำที่ใช้แทนกระดาษก่อน โดยหน่วยความจำจะมีลักษณะเหมือนตารางมีขนาด 600 x 120 ไบท์ โดยแต่ละไบท์จะใช้ 6 บิท แต่ละบิทจะแทนจุดจุดหนึ่งบนกระดาษ ถ้ามีค่าเป็นศูนย์ แสดงว่า ไม่มีการพลอตที่จุดนั้น ถ้ามีค่าเป็นหนึ่งแสดงว่ามีการพลอตที่จุดนั้น

คำสั่งที่เก็บไว้อยู่ในลักษณะของเวกเตอร์คือเส้นตรง เพราะฉะนั้นเมื่อจะทำการพลอตลงบนหน่วยความจำ จะต้องทำการคำนวณหาจุดที่จะพลอต เพื่อเป็นเส้นตรงของเวกเตอร์นี้ (ดังที่ได้อธิบายในโปรแกรมย่อย FDRWLINE) หลังจากพลอตทุกคำสั่งเรียบร้อยแล้วก็จะเริ่มนำข้อมูลในหน่วยความจำเก็บในแฟ้มข้อมูลก่อน โดยแต่ละระเบียน (RECORD) ในแฟ้มข้อมูลจะมี รหัสควบคุมให้เครื่องพิมพ์ทำงานแบบกราฟฟิก (คือรหัส 05) จะนำข้อมูล 1 ไบท์ อยู่ต้นของระเบียน เพื่อควบคุม 6 บิทในไบท์ต่อๆมา พลอตจนครบแล้วทำการพลอตข้อมูลของระเบียนต่อไป อ่านข้อมูลจนหมดแฟ้มข้อมูล เพราะฉะนั้น โดยสรุปคือ

1. เก็บคำสั่งพลอตไว้ในแฟ้มข้อมูล
2. เมื่อมีความต้องการพลอตลงกระดาษ จะนำแฟ้มข้อมูลที่เก็บคำสั่งพลอตออกมาทำการพลอตลงหน่วยความจำที่ใช้แทนกระดาษ โดยใช้โปรแกรมย่อย BDRWLINE
3. นำข้อมูลในหน่วยความจำที่ใช้แทนกระดาษ เก็บลงในแฟ้มข้อมูล
4. ส่งแฟ้มข้อมูลออกพิมพ์ ในลักษณะเช่นเดียวกับแฟ้มข้อมูลทั่วไป (SPOOL filename)



BINARY	OCT	DEC	HEX	ASCII	2 4 6 8 10 12 / 1 3 5 7 9 11	BINARY	OCT	DEC	HEX	ASCII	2 4 6 8 10 12 / 1 3 5 7 9 11	BINARY	OCT	DEC	HEX	ASCII	2 4 6 8 10 12 / 1 3 5 7 9 11
0100000	040	32	20	Space		1000000	100	64	40	@		1100000	140	96	60	`	
0100001	041	33	21	!		1000001	101	65	41	A		1100001	141	97	61	a	
0100010	042	34	22	"		1000010	102	66	42	B		1100010	142	98	62	b	
0100011	043	35	23	#		1000011	103	67	43	C		1100011	143	99	63	c	
0100100	044	36	24	$		1000100	104	68	44	D		1100100	144	100	64	d	
0100101	045	37	25	%		1000101	105	69	45	E		1100101	145	101	65	e	
0100110	046	38	26	&		1000110	106	70	46	F		1100110	146	102	66	f	
0100111	047	39	27	'		1000111	107	71	47	G		1100111	147	103	67	g	
0101000	050	40	28	(		1001000	110	72	48	H		1101000	150	104	68	h	
0101001	051	41	29	)		1001001	111	73	49	I		1101001	151	105	69	i	
0101010	052	42	2A	*		1001010	112	74	4A	J		1101010	152	106	6A	j	
0101011	053	43	2B	+		1001011	113	75	4B	K		1101011	153	107	6B	k	
0101100	054	44	2C	,		1001100	114	76	4C	L		1101100	154	108	6C	l	
0101101	055	45	2D	-		1001101	115	77	4D	M		1101101	155	109	6D	m	
0101110	056	46	2E	.		1001110	116	78	4E	N		1101110	156	110	6E	n	
0101111	057	47	2F	/		1001111	117	79	4F	O		1101111	157	111	6F	o	
0110000	060	48	30	0		1010000	120	80	50	P		1110000	160	112	70	p	
0110001	061	49	31	1		1010001	121	81	51	Q		1110001	161	113	71	q	
0110010	062	50	32	2		1010010	122	82	52	R		1110010	162	114	72	r	
0110011	063	51	33	3		1010011	123	83	53	S		1110011	163	115	73	s	
0110100	064	52	34	4		1010100	124	84	54	T		1110100	164	116	74	t	
0110101	065	53	35	5		1010101	125	85	55	U		1110101	165	117	75	u	
0110110	066	54	36	6		1010110	126	86	56	V		1110110	166	118	76	v	
0110111	067	55	37	7		1010111	127	87	57	W		1110111	167	119	77	w	
0111000	070	56	38	8		1011000	130	88	58	X		1111000	170	120	78	x	
0111001	071	57	39	9		1011001	131	89	59	Y		1111001	171	121	79	y	
0111010	072	58	3A	:		1011010	132	90	5A	Z		1111010	172	122	7A	z	
0111011	073	59	3B	;		1011011	133	91	5B	[		1111011	173	123	7B	{	
0111100	074	60	3C	<		1011100	134	92	5C	\		1111100	174	124	7C	¦	
0111101	075	61	3D	=		1011101	135	93	5D	]		1111101	175	125	7D	}	
0111110	076	62	3E	>		1011110	136	94	5E	^		1111110	176	126	7E	~	
0111111	077	63	3F	?		1011111	137	95	5F	—		1111111	177	127	7F	Delete	

รูปที่ ข-1 ตารางรหัสข้อมูลควบคุมเครื่องพิมพ์

ข-2.1 จอภาพ

จอภาพที่ใช้เป็นผลิตภัณฑ์ที่มีเครื่องหมายการค้าว่า PT200 และมีวงจรเพิ่มเติมสำหรับทำหน้าที่ทางกราฟฟิก มีโหมดกราฟฟิก 2 โหมด คือ โหมดแรกเป็นกราฟฟิกพื้นฐานมีจำนวนจุดคือ 320x720 จุด อีกโหมดมีสภาพเหมือนจอภาพกราฟฟิก TEKTRONIX 4014 สามารถรับคำสั่งได้เหมือนกัน ซึ่งในโหมดนี้ ใช้ในการพลอตโดยโปรแกรมย่อยทางกราฟฟิกในโหมดนี้จะเสมือนมีจำนวนจุดอยู่ 780x1024 จุด (โปรดดูในคู่มือการใช้จอภาพ[๑,๑๐])

ข-2.2 การสร้างโปรแกรมควบคุมการแสดงผลลัพธ์ทางจอภาพ

เนื่องจากจอภาพเป็นอุปกรณ์แสดงผลลัพธ์ได้ในทันที เมื่อมีคำสั่งพลอตเข้ามา คำสั่งนี้จะถูกแปลงออกเป็น คำสั่งพลอตของอุปกรณ์แสดงผลลัพธ์จำลอง (โปรดดูคำอธิบายอุปกรณ์แสดงผลลัพธ์จำลองได้จาก บทย่อยที่ 2.1) ซึ่งอยู่ในรูปของตำแหน่งแนวราบ ตำแหน่งแนวดิ่ง และคำสั่งยกหรือลากปากกา เมื่อต้องการนำคำสั่งพลอตของอุปกรณ์แสดงผลลัพธ์จำลองนี้ออกแสดงทางจอภาพ สิ่งที่ต้องทำคือ

- การแปลงตำแหน่งของอุปกรณ์จำลองเป็นตำแหน่งจริงบนจอภาพ
- ทำตามคำสั่งลากเส้น หรือย้ายตำแหน่งพลอตบนจอภาพ

ในกรณีที่มีแต่การย้ายตำแหน่งจะไม่ส่งคำสั่งใดไปยังจอภาพ เพียงแต่เปลี่ยนค่าที่จำตำแหน่งลุดท้ายที่มีการพลอตบนจอภาพเป็นค่าของตำแหน่งนี้ แต่ถ้าเป็นคำสั่งที่มีการลากเส้นจะส่งตำแหน่งลุดท้ายที่มีการพลอตบนจอภาพ และตำแหน่งของคำสั่งนี้ซึ่งจะเป็นจุดปลายของเส้นตรงที่ต้องการลากไปยังจอภาพ โดยก่อนส่งจะต้องทำการแปลงตำแหน่งทั้งลองให้เป็นตำแหน่งจริงบนจอภาพก่อน เมื่อส่งตำแหน่งทั้งลองให้จอภาพแล้วก็จะได้เส้นตรงตามต้องการ เพราะฉะนั้นโดยสรุปคือ

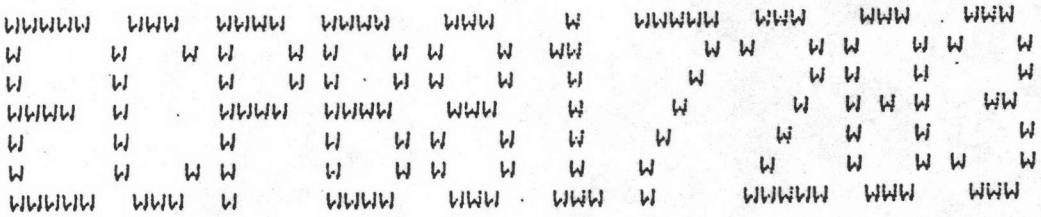
1. รับคำสั่งพลอตเข้ามายังอุปกรณ์แสดงผลลัพธ์จำลอง
2. ถ้าคำสั่งนี้เป็นคำสั่งที่ย้ายตำแหน่งที่ทำการพลอตครั้งลุดท้าย ก็จะเพียงแต่เปลี่ยนค่าที่จำตำแหน่งลุดท้ายที่มีการพลอตเท่านั้น จะไม่มีการติดต่อกับจอภาพจริง
3. แต่ถ้าคำสั่งนี้เป็นคำสั่งในการลากเส้น ก็จะส่งตำแหน่งที่ทำการพลอตครั้งสุดท้าย และตำแหน่งล่าสุดที่ส่งมาพร้อมกับคำสั่งนี้ไปยังจอภาพ เพื่อทำการลากเส้นบนจอภาพ
4. หลังจากนี้ก็จะรอรับคำสั่งต่อไป

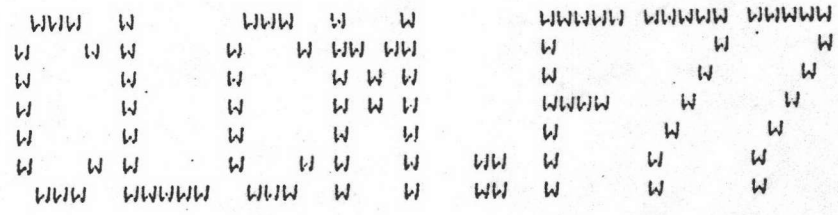
ภาคผนวก ค.

รายละเอียดโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>CLCM.F77





Label: PRT007 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>CLCM.F77
File last modified: 87-09-08.16:59:40.Tue

Spooled: 87-09-18.10:37:20.Fri [Spooler rev 19.4],
Started: 87-09-18.10:43:04.Fri on: PRO by: PRO

```
C
C
C
C     BASIC GRAPHIC SUBROUTINE PACKAGE
C
C
C     CONTENTS:
C
CX          AXIS    -- PLOT A LABELED AXIS
CX          ANGLE   -- SET ANGLE
CX          COSD    -- RETURN COS(X), WHERE X IS IN DEGREES
CX          CSLANT  -- SET CHARACTER SLANT
CX          DASHLN  -- PLOT DOTTED/DASHED VECTOR
CX          FACTOR  -- SET SCALING FACTOR
CX          LINE    -- PLOT LINES FROM TWO ARRAYS
CX          LINEWT  -- SET LINE WEIGHTING FACTOR
CX          RCTNGL  -- PLOT A RECTANGLE
CX          SCALE   -- SCALE AN ARRAY FOR A GIVEN LENGTH
CX          SIND    -- RETURN SIN(X), WHERE X IS IN DEGREES
CX          WHERE   -- RETURN CURRENT PEN POSITION
C
C
C
C     PLOT LINES FROM TWO ARRAYS
C
      SUBROUTINE LINE(X,Y,NPAIRS,INC,SYMCTL,SYM)

      INTEGER*4   NPAIRS,INC,SYMCTL
      INTEGER*4   SYM
      REAL*4      X(1),Y(1)

C
C     PARAMETERS :
C                 X,Y       ARE THE 2 ARRAYS TO BE PLOTTED
C                 NPAIRS    # PAIRS OF DATA POINTS TO BE PLOTTED
C                 INC       THE INCREMENT OF THE SUBSCRIPT FOR EACH CONSECUTIVE
C                           ELEMENT TO BE PLOTTED
C                 SYMCTL    LINE PLOTTING CONTROL VARIABLE:
C                           =0 FOR LINE ONLY
C                           <0 SYMBOL DRAWN AT EACH IABS(SYMCTL)'TH POINT
C                              NO LINE IS DRAWN
C                           >0 SYMBOL DRAWN AT EACH IABS(SYMCTL)'TH POINT
C                              A LINE IS DRAWN FROM PREVIOUS POSITION
C                 SYM       # OF THE SYMBOL TO BE DRAWN  (0-9)
C

      INTEGER*4   I,IEND,INCX,IPEN,J,IDUMMY
      REAL*4      SX,SY,PSX,PSY
      REAL*4      XMIN,XSCALE,YMIN,YSCALE,XSTART,YSTART


c      write(15,*)'line',x,y,inc,symctl,SYM


      IEND=NPAIRS*INC                          /* TERMINATE PARAM FOR LOOP

C
C     FETCH PARAMETERS GENERATED BY SCALE

      XMIN=X(IEND+1)
      YMIN=Y(IEND+1)
```

```
C
      J=IEND+INC+1
      XSCALE=X(J)
      YSCALE=Y(J)
c      write(15,*)'line',xmin,ymin,xscale,yscale,J

      INCX=INC
      IF(SYMCTL.LT.0)INCX=INC*IABS(SYMCTL)

C
C     DRAW LINE AND/OR SYMBOLS

      PSX = 0.
      PSY = 0.
c      write(15,*)'line iend=',iend
      DO 50 I=1,IEND,INCX

c        write(15,*)'line i',i,XSCALE
        SX = (X(I)-XMIN)/XSCALE          /* SCALED X
        SY = (Y(I)-YMIN)/YSCALE          /* SCALED Y
c        write(15,*)'line SXY',SYMCTL

        IF (SYMCTL.LT.0) GOTO 5                  /* DRAW SYMBOL ONLY
c        write(15,*)'line *1'
c        IPEN=2
c        IF   (I.EQ.1)   IPEN = 3
        IF   (I.EQ.1)   PSX = SX
        IF   (I.EQ.1)   PSY = SY
c        write(15,*)'line *2',PSX,PSY,SX,SY,IABS(SYMCTL)
        CALL DASHLN(PSX,PSY,SX,SY,IABS(SYMCTL))
        PSX = SX
        PSY = SY
c        write(15,*)'line *3',IPEN

C        call plot(SX,SY,IPEN)
        IF (SYMCTL.EQ.0) GOTO 50                 /* LINE ONLY
c        write(15,*)'line *4'
        J=I-1
        IF (SYMCTL.LE.0) GOTO 50
5       CONTINUE
c        write(15,*)'line XY=*',SX,SY,SYM
        CALL SYMBOL(SX-.13,SY-.13,.13,IDUMMY,SYM,0.,-1)
c        write(15,*)'line XY=**',SX,SY,iend,incx

50    CONTINUE


9999  RETURN
      END

C
C     SCALE AN ARRAY FOR A GIVEN LENGTH
C
      SUBROUTINE SCALE(ARRAY,AXLEN,NPTS,INC)

      INTEGER*4   NPTS,INC
      REAL*4      ARRAY(1),AXLEN

C
C     PARAMETERS :
C                 ARRAY     IS THE ARRAY TO BE SCALED
```

```
C
C
C                AXLEN     IS THE AXIS LENGTH FOR WHICH IT IS TO BE SCALED
C                NPTS      # POINTS TO BE SCALED
C                INC       THE INCREMENT OF THE SUBSCRIPT FOR EACH ELEMENT
C                          TO BE EXAMINED
C
C     THIS SUBROUTINE DOES NOT ALTER THE ORIGINAL DATA IN ARRAY, BUT
C     CREATES 2 ADDITIONAL ELEMENTS:
C
C     AT (NPTS*INC)+1   -   SMALLEST VALUE IN ARRAY
C     AT (NPTS*INC)+INC+1   -   DELTA-VALUE OF ARRAY PER PLOTTED INCH
C
C     NOTE - THESE EXTRA ELEMENTS MUST BE DIMENSIONED FOR, ELSE CATASTROPHE
C          WILL OCCUR
C
      INTEGER*4 I,IEND,ISTART
      REAL*4    RHI,RLO

      RLO=ARRAY(1)                                  /* GET INITIAL LOW, HIGH
      RHI=ARRAY(1)

      ISTART=INC+1                                  /* STARTING ELEMENT #
      IEND=NPTS*INC                                 /* TERMINATING ELEMENT #

      DO 20 I=ISTART,IEND,INC
       IF(ARRAY(I).LT.RLO)RLO=ARRAY(I)
       IF(ARRAY(I).GT.RHI)RHI=ARRAY(I)
20    CONTINUE

      I=IEND)+1
      ARRAY(I)=RLO                                  /* 1ST XTRA ELEMENT - LEAST ELEMEN
      I=I+INC
      ARRAY(I)=ABS(RHI-RLO)/AXLEN                   /* 2ND XTRA ELEMENT - DELTA-VALUE

      RETURN
      END


C
C     PRODUCE BAR CHART
C
      SUBROUTINE BAR(X,Y,ARRAY,NBAR,XLENG)

      REAL*4      X,Y,XLENG,ARRAY(1)
      INTEGER*4   NBAR

      REAL*4      BLK,XTEMP,YTEMP

      IF ((NBAR.LT.0).OR.(NBAR.GT.99)) RETURN /* ERROR AT NUMBER OF BAR

      BLK = XLENG/REAL(NBAR)
      XTEMP = BLK/2.
      DO 100 J = 1,NBAR

        CALL PLOT(X+XTEMP,Y,3)
        CALL RPLOT(0.,ARRAY(J),2)
        CALL RPLOT((BLK/2.),0.,2)
        CALL RPLOT(0.,-ARRAY(J),2)
        XTEMP = XTEMP + BLK
```

```
C

100   CONTINUE

999   CONTINUE
      RETURN
      END

C
C     PRODUCE BAR CHART
C
      SUBROUTINE BAR2(X,Y,ARRAY1,ARRAY2,NBAR,XLENG)

      REAL*4      X,Y,XLENG,ARRAY1(1),ARRAY2(1)
      INTEGER*4   NBAR

      REAL*4      BLK,XTEMP,YTEMP

      IF ((NBAR.LT.0).OR.(NBAR.GT.99)) RETURN /* ERROR AT NUMBER OF BAR

      BLK = XLENG/REAL(NBAR)
      XTEMP = BLK/3.
      DO 100 J = 1,NBAR

        CALL PLOT(X+XTEMP,Y,3)
        CALL RPLOT(0.,ARRAY1(J),2)
        CALL RPLOT((BLK/3.),0.,2)
        CALL RPLOT(0.,-ARRAY1(J),2)

        CALL PLOT(X+XTEMP+(BLK/3.),Y,3)
        CALL RPLOT(0.,ARRAY2(J),2)
        CALL RPLOT((BLK/3.),0.,2)
        CALL RPLOT(0.,-ARRAY2(J),2)

        XTEMP = XTEMP + BLK

100   CONTINUE

999   CONTINUE
      RETURN
      END)


C
C     PRODUCE PIE CHART
C
      SUBROUTINE PIE(XC,YC,ARRAY,NPIE,RD,MODE)

      REAL*4      XC,YC,RD,ARRAY(1)
      INTEGER*4   NPIE,MODE

      IF ((MODE.NE.1).AND.(MODE.NE.2)) RETURN
      IF (MODE.EQ.1) CALL PIE1(XC,YC,ARRAY,NPIE,RD)
      IF (MODE.EQ.2) CALL PIE2(XC,YC,ARRAY,NPIE,RD)


      RETURN
      END

C
C     PRODUCE ONE PIECE PIE CHART
C
```

```
      SUBROUTINE PIE1(XC,YC,ARRAY,NPIE,RD)

      REAL*4      XC,YC,RD,ARRAY(1)
      INTEGER*4   NPIE

      REAL*4      TOTAL,ATEMP,TEMP1,TEMP2,XTEMP,YTEMP
      INTEGER*4   J,INTEG

      IF ((NPIE.LT.0).OR.(NPIE.GT.19)) RETURN

      TOTAL = 0.
      DO 100 J = 1,NPIE
        TOTAL = TOTAL+ARRAY(J)
100   CONTINUE
      CALL CIRCLE(XC+RD,YC,XC,YC,0.,1)

      CALL PLOT(XC,YC,3)
      CALL RPLOT(RD,0.,2)
      ATEMP = 0.
      DO 200 J = 1,NPIE-1
        ATEMP = ATEMP+360.*ARRAY(J)/TOTAL
        CALL PLOT(XC,YC,3)
        CALL RPLOT(RD*COSD(ATEMP),RD*SIND(ATEMP),2)

200   CONTINUE

      ATEMP = 0.
      DO 300 J = 1,NPIE
        TEMP1 = 360.*ARRAY(J)/TOTAL
        XTEMP = XC + (RD*1.1)*COSD(ATEMP+(TEMP1/2.))
        IF ( ((ATEMP+(TEMP1/2.)).GT.90. ) .AND.
     X       ((ATEMP+(TEMP1/2.)).LT.270.)       )
     X     XTEMP = XTEMP-(9.*RD/REAL(NPIE*1.5))
        YTEMP = YC + (RD*1.1)*SIND(ATEMP+(TEMP1/2.))
        TEMP2 = ARRAY(J)*100/TOTAL
        CALL SYMBOL(XTEMP,YTEMP,RD/REAL(NPIE*1.5),'(',INTEG,0.,1)
        CALL NUMBER(999.,999.,RD/REAL(NPIE*1.5),REAL(J),0.,-1)
        CALL SYMBOL(999.,999.,RD/REAL(NPIE*1.5),')',INTEG,0.,1)
        CALL NUMBER(999.,999.,RD/REAL(NPIE*1.5),TEMP2,0.,2)
        CALL SYMBOL(999.,999.,RD/REAL(NPIE*1.5),'%',INTEG,0.,1)
        ATEMP = ATEMP+TEMP1
300   CONTINUE

9999  CONTINUE
      RETURN
      END
C
C     PRODUCE MANY PIECES PIE CHART
C
      SUBROUTINE PIE2(XC,YC,ARRAY,NPIE,RD)

      REAL*4      XC,YC,RD,ARRAY(1)
      INTEGER*4   NPIE

      REAL*4      TOTAL,ATEMP,TEMP1,TEMP2,XTEMP,YTEMP
      REAL*4      XTEMP1,YTEMP1
      INTEGER*4   J,INTEG

      IF ((NPIE.LT.0).OR.(NPIE.GT.19)) RETURN
```

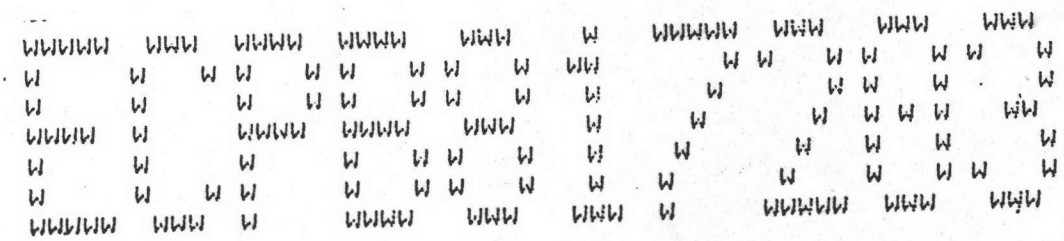
```
C
      TOTAL = 0.
      DO 100 J = 1,NPIE
        TOTAL = TOTAL+ARRAY(J)
100   CONTINUE

      ATEMP = 0.
      DO 400 J = 1,NPIE
        TEMP1 = 360.*ARRAY(J)/TOTAL
        XTEMP = XC + (RD*.1)*COSD(ATEMP+(TEMP1/2.))
        YTEMP = YC + (RD*.1)*SIND(ATEMP+(TEMP1/2.))
        XTEMP1 = XTEMP + (RD*.9)*COSD(ATEMP)
        YTEMP1 = YTEMP + (RD*.9)*SIND(ATEMP)
        CALL PLOT(XTEMP,YTEMP,3)
        CALL PLOT(XTEMP1,YTEMP1,2)
        CALL CIRCLE(XTEMP,YTEMP,ATEMP,ATEMP+TEMP1,RD*.9,7)
        XTEMP1 = XTEMP + (RD*.9)*COSD(ATEMP+TEMP1)
        YTEMP1 = YTEMP + (RD*.9)*SIND(ATEMP+TEMP1)
        CALL PLOT(XTEMP,YTEMP,3)
        CALL PLOT(XTEMP1,YTEMP1,2)

        ATEMP = ATEMP+TEMP1
400   CONTINUE


      ATEMP = 0.
      DO 300 J = 1,NPIE
        TEMP1 = 360.*ARRAY(J)/TOTAL
        XTEMP = XC + (RD*1.1)*COSD(ATEMP+(TEMP1/2.))
        IF ( ((ATEMP+(TEMP1/2.)).GT.90.  ) .AND.
     X       ((ATEMP+(TEMP1/2.)).LT.270.)       )
     X     XTEMP = XTEMP-(9.*RD/REAL(NPIE*1.5))
        YTEMP = YC + (RD*1.1)*SIND(ATEMP+(TEMP1/2.))
        TEMP2 = ARRAY(J)*100/TOTAL
        CALL SYMBOL(XTEMP,YTEMP,RD/REAL(NPIE*1.5),'(',INTEG,0.,1)
        CALL NUMBER(999.,999.,RD/REAL(NPIE*1.5),REAL(J),0.,-1)
        CALL SYMBOL(999.,999.,RD/REAL(NPIE*1.5),')',INTEG,0.,1)
        CALL NUMBER(999.,999.,RD/REAL(NPIE*1.5),TEMP2,0.,2)
        CALL SYMBOL(999.,999.,RD/REAL(NPIE*1.5),'%',INTEG,0.,1)
        ATEMP = ATEMP+TEMP1
300   CONTINUE

9999  CONTINUE
      RETURN
      END

C
C     PRODUCE ARROW
C
      SUBROUTINE AROHD(X,Y,XTIP,YTIP,AHLEN,AHWD,ITYPE)

      REAL*4      X,Y,XTIP,YTIP,AHLEN,AHWD
      INTEGER*4   ITYPE

      REAL*4      ANGLE

      IF ((X.EQ.XTIP).AND.(Y.EQ.YTIP)) GOTO 9999
      XLEN = (ABS(X-XTIP)**2.+ABS(Y-YTIP)**2.)**.5
      IF (AHLEN.GT.XLEN) GOTO 900

      IF ((ITYPE.LT.1).OR.(ITYPE.GT.3)) ITYPE = 1
```

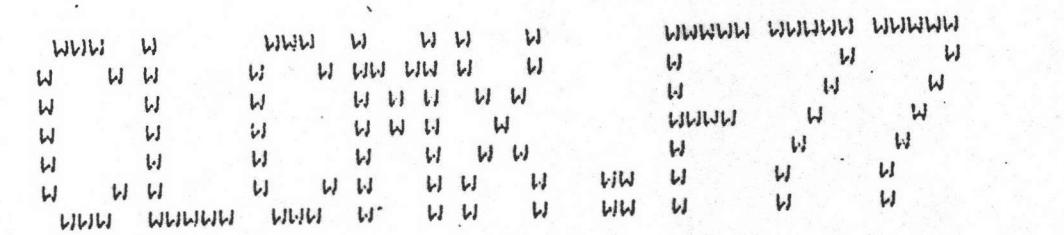
```
C

      CALL DEGREE(X,Y,XTIP,YTIP,ANGLE)

      IF (ITYPE.EQ.2) GOTO 500
      IF (JTYPE.EQ.3) GOTO 700

      CALL PLOT(X,Y,3)
      CALL PLOT(XTIP,YTIP,2)
      CALL ROTATE(ANGLE)
      CALL RELPLT(-AHLEN,AHWD/2.,2)
      CALL RELPLT(AHLEN,-AHWD/2.,2)
      CALL RELPLT(-AHLEN,-AHWD/2.,2)
      CALL RELPLT(AHLEN,AHWD/2.,2)
      CALL ROTATE(0.)
      GOTO 9999

500   CONTINUE
      CALL PLOT(X,Y,3)
      CALL PLOT(XTIP,YTIP,2)
      CALL ROTATE(ANGLE)
      CALL RELPLT(-AHLEN,AHWD/2.,2)
      CALL RELPLT(0.,-AHWD,2)
      CALL RELPLT(AHLEN,AHWD/2.,2)
      CALL ROTATE(0.)
      GOTO 9999

700   CONTINUE
      CALL PLOT(X,Y,3)
C      write(15,*)xtip,XTIP-AHLEN*COSD(ANGLE),
C     *           ytip,YTIP-AHLEN*SIND(ANGLE)
      CALL PLOT(XTIP-AHLEN*COSD(ANGLE),YTIP-AHLEN*SIND(ANGLE),2)
      CALL ROTATE(ANGLE)
      CALL RELPLT(0.,-AHWD/2.,2)
      CALL RELPLT(AHLEN,AHWD/2.,2)
      CALL RELPLT(-AHLEN,AHWD/2.,2)
      CALL RELPLT(0.,-AHWD/2.,2)
      CALL ROTATE(0.)
      GOTO 9999

700   CONTINUE
      CALL PLOT(X,Y,3)
      CALL PLOT(XTIP,YTIP,2)
      GOTO 9999

9999  CONTINUE
      RETURN
      END
C
C     BLACK RECTANGULA
C
      SUBROUTINE RCTBLK (X,Y,H,W,THETA)

      REAL*4  X,Y,H,W,THETA

$INSERT LGDEVICE.F77

      CALL RCTNGL(X,Y,H,W,THETA,3)
      NUMLINE = INTL(H*RPIY*SFACTR)
C      write(15,*)numline,H,RPIY,SFACTR
      YINC    = 0.
      IF (NUMLINE.NE.0) YINC = H/REAL(NUMLINE)
```

```
C
      CALL ROTATE(THETA)
      CALL PLOT(X,Y,3)
      DO 100 J=1,NUMLINE
        CALL RELPLT(0.,YINC,3)
        CALL RELPLT(W,0.,2)
        CALL RELPLT(-W,0.,3)
100   CONTINUE
      CALL ROTATE(0.)

      RETURN
      END

C
C     CALCULATE DEGREE OF LINE SEGMENT
C
      SUBROUTINE DEGREE (X1,Y1,X2,Y2,ANGLE)

      REAL*4      X1,Y1,X2,Y2,ANGLE

      IF ((X1.EQ.X2).AND.(Y1.EQ.Y2)) GOTO 500
      IF (X1.EQ.X2) GOTO 600
      IF (Y1.EQ.Y2) GOTO 700

      ANGLE = ATAN((Y2-Y1)/(X2-X1))*57.29578
      IF ( ((Y2-Y1).GT.0).AND.((X2-X1).LT.0) ) ANGLE = 180.  + ANGLE
      IF ( ((Y2-Y1).LT.0).AND.((X2-X1).LT.0) ) ANGLE = -180.  + ANGLE
      GOTO 9999

500   CONTINUE
      ANGLE = 0.
      GOTO 9999

600   CONTINUE
      ANGLE = 90.
      IF (Y2.LT.Y1) ANGLE = -90.
      GOTO 9999

700   CONTINUE
      ANGLE = 0.
      IF (X2.LT.X1) ANGLE = 180.

9999  CONTINUE
      RETURN
      END
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>CLCMX.F77





Label: PRT008 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>CLCMX.F77
File last modified: 87-09-04.12:47:12.Fri

Spooled: 87-09-18.10:37:24.Fri [Spooler rev 19.4]
Started: 87-09-18.10:44:32.Fri on: PRO by: PRO

```
C
C
C
C
      SUBROUTINE NEWPEN(ID)
      INTEGER*4  ID

      RETURN
      END

C
C
C
      SUBROUTINE TLRNCE(ID1,ID2)

      INTEGER*4  ID1,ID2

      RETURN
      END

C
C
C
      SUBROUTINE FONT(D)

      REAL*4     D

      RETURN
      END

C
C
C
      SUBROUTINE SETCHR(D1,D2,D3,D4,D5)

      REAL*4     D1,D2,D3,D4,D5

      RETURN
      END

C
C
C
      SUBROUTINE PRFORM(ID1,ID2,ID3)

      INTEGER*4  ID1,ID2,ID3

      RETURN
      END

C
C
C
      SUBROUTINE WINDOW(ID1,ID2,ID2,ID3,ID4,ID5)

      INTEGER*4  ID1,ID2,ID3,ID4,ID5

      RETURN
      END

C
```

```
C
C
C
      SUBROUTINE DASHS(D,ID)

      INTEGER*4   ID
      REAL*4      D

      RETURN
      END

C
C
C
      SUBROUTINE NEWPLT

      INTEGER*2   INKEY

      CALL ENDPLT
      WRITE(1,*)'WOULD YOU LIKE TO HARDCOPY THIS PICTURE (Y/N) ?'
100   CALL TTIN(INKEY)
      IF (CHAR(INKEY).EQ.'Y') CALL HARDCOPY
      IF (CHAR(INKEY).EQ.'y') CALL HARDCOPY
      IF ( (CHAR(INKEY).NE.'Y').AND.(CHAR(INKEY).NE.'y').AND.
     X     (CHAR(INKEY).NE.'N').AND.(CHAR(INKEY).NE.'n') ) GOTO 100
      CALL INITPARA

      RETURN
      END

C
C
C
      SUBROUTINE MIRROR(ITYPE)

      INTEGER*4   ITYPE

$INSERT LGDEVICE.F77

      IF ((ITYPE.LT.0).OR.(ITYPE.GT.3)) ITYPE=0
      MIRROR = INTS(ITYPE)

      RETURN
      END

C
C
C
      SUBROUTINE AXIS(X,Y,IARR,NCHAR,XL,ANG,FV,DV)

      REAL*4      X,Y,XL,ANG,FV,DV
      INTEGER*4   NCHAR
      INTEGER*2   IARR(1)

      CALL XAXIS(X,Y,IARR,NCHAR,XL,ANG,FV,DV,DV,0)

      RETURN
      END

C
C     OPERATION MESSAGE
```

```
C


C
      SUBROUTINE OPNES(NCT,XMSG)

      INTEGER*4  NCT,I
      INTEGER*2  XMSG(1)
      CHARACTER*1 INKEY

$INSERT LSDEVICE.F77

      IF (NCT.LT.0) GOTO 1000
C     OTHER NCT .GE. 0
      IF (.NOT.FILEPLOT)   CALL ENTNAT
      IF (.NOT.FILEPLOT)   CALL SELSCR(INTS(0))
      IF (.NOT.FILEPLOT)   CALL ENTTEK
      IF (.NOT.FILEPLOT)   CALL ENTTXT
      DO 1020 J = 1,30
      WRITE(1,*)
1020  CONTINUE
      CALL TNOUA(XMSG,INTS(NCT))
      WRITE(1,*)
      WRITE(1,*)
      WRITE(1,*)'Please Press Return KEY to continue'
      READ(1,5)INKEY
5     FORMAT(A1)
      IF (.NOT.FILEPLOT)   CALL ENTGRP
      IF (.NOT.FILEPLOT)   CALL ENTNAT
      IF (.NOT.FILEPLOT)   CALL SELSCR(INTS(0))
      IF (.NOT.FILEPLOT)   CALL ENTTEK
      GOTO 9999

1000  CONTINUE
      I = IABS(NCT)
      IF (.NOT.FILEPLOT)   CALL ENTTXT
      DO 1010 J = 1,30
      WRITE(1,*)
1010  CONTINUE
      CALL TNOUA(XMSG,INTS(I))
      WRITE(1,*)
      WRITE(1,*)
      WRITE(1,*)'Please Press Return KEY to continue'
      READ(1,5)INKEY
      IF (.NOT.FILEPLOT)   CALL ENTGRP
      GOTO 9999

9999  CONTINUE
      RETURN
      END
C
C     SET LINE WEIGHTING FACTOR
C
      SUBROUTINE LINEWT(LW)

      INTEGER*4  LW

C
C     PARAMETERS :
C                 LW  =  LINE WEIGHT, FROM 1 TO 8
C
$INSERT LSDEVICE.F77
```

```
C:



      IF(LW.LT.1.OR.LW.GT.8)GO TO 10
      LNWT=INTS(LW)
      GOTO 9999

C
C   ERROR...

10    WRITE(ERRDEV,11)LW
11    FORMAT(' BAD CALL TO LINEWT, LW=',I6)

9999  RETURN
      END

C
C     PLOT A LABELED AXIS
C
      SUBROUTINE XAXIS(X,Y,TITLE,NCH,AXLEN,THETA,FIRSTV,DELTAV,DELTAT,
     +                 NMIN)

      INTEGER*2  TITLE(1)
      INTEGER*4  NCH,NMIN
      REAL*4     X,Y,AXLEN,THETA,FIRSTV,DELTAV,DELTAT

C
C     PARAMETERS :
C                X,Y      DEFINE THE START OF THE AXIS (IN INCHES)
C                TITLE    IS THE TITLE, IF ANY, IN A2 FORMAT
C                NCH      THE LENGTH OF THE TITLE, IN CHARACTERS
C                AXLEN    AXIS LENGTH  (IN INCHES)
C                THETA    ANGLE OF ROTATION OF THE AXIS, FROM THE +X AXIS
C                FIRSTV*  MINIMUM VALUE IN THE ARRAY TO BE PLOTTED
C                DELTAV*  CHANGE IN VALUE OF THE ARRAY FOR EACH INCH OF THE AXIS
C                DELTAT   CHANGE IN VALUE OF THE ARRAY BETWEEN PLOTTED TIC MARKS
C                NMIN     # MINOR TIC MARKS BETWEEN MAJOR ONES
C
C      NOTE:     A * DENOTES THAT THESE VALUES MAY BE GENERATED BY SUBROUTINE
C                'SCALE'
C
$INSERT LGDEVICE.F77


      INTEGER*4  I,ICH,LNWT1,NDEC,IDUMMY
      REAL*4     DISMAJ,DISMIN,HWIDE,RFAC,RX,RY,TMAJLN,TMINLN,VAL
      REAL*4     XDISP,XLIMIT,XLOCN,XMIN,YDISP,FDUMMY
      LOGICAL    PUTDEC

      CALL PLOT(X,Y,3)                          /* MOVE TO START OF AXIS
      TMINLN=.1*FLOAT(ISIGN(1,NCH))             /* MINOR TIC LENGTH (& DIRECTION)
      TMAJLN=.2*FLOAT(ISIGN(1,NCH))             /* MAJOR TIC LENGTH
      TSIN=SIND(THETA)                          /* GENERATE SIN, COS (THETA)
      TCOS=COSD(THETA)                          /* FOR INTERNAL ROTATION

C
C   DRAW AXIS VECTOR

      LNWT1=LNWT                                /* SAVE LINE WT
      LNWT=4
      CALL RELPLT(AXLEN,0.,2)
      CALL PLOT(X,Y,3)                          /* AND GO HOME AGAIN
```

```
C
      DISMAJ=DELTAT/DELTAV                          /*'DIST BETWEEN MAJOR TIC MARKS
      XLIMIT=AXLEN+X                                /* REAL END OF AXIS
      DISMIN=DISMAJ                                 /* DEFAULT (FOR NO MINOR TIC MARKS)
      IF(NMIN.LE.0)GO TO 5                          /* BRANCH IF NO MINNOR TIC MARKS
      XMIN=NMIN+1
      DISMIN=DISMAJ/XMIN                            /* DISTANCE BETWEEN MINOR TIC MARKS

5     XLOCN=X                                       /* CURRENT 'X' LOCATION

C
C  PLOT MAJOR TIC MARK

10    LNWT=3
      CALL RELPLT(0.,TMAJLN,2)
      CALL RELPLT(0.,-TMAJLN,3)
      IF(NMIN.LE.0)GO TO 20

C
C   PLOT MINOR TIC MARKS

      DO 15 I=1,NMIN
      XLOCN=XLOCN+DISMIN                            /* UPDATE X LOCATION
      IF(XLOCN.GE.XLIMIT)GO TO 100                  /* REACHED END OF AXIS
      CALL PLOT(X,Y,3)                              /* ACCOUNT FOR ACCURACY LOSS
      CALL RELPLT(XLOCN-X,0.,3)
      LNWT=1                                        /* MINOR TIC MARKS 1 RASTER WIDE
15    CALL RELPLT(0.,TMINLN,2)

20    XLOCN=XLOCN+DISMIN                            /* UPDATE LOC'N
      IF(XLOCN.GT.XLIMIT)GO TO 100                  /* BRANCH IF WE'VE REACHED THE END
      CALL PLOT(X,Y,3)
      CALL RELPLT(XLOCN-X,0.,3)
      GO TO 10

C
C  LABEL MAJOR TIC MARKS

100   YDISP=-.4                                     /* 'Y' DISPLACEMENT
      IF(NCH.GE.0)YDISP=.33
      XDISP=0.                                      /* CURRENT X-DISPLACEMENT
      LNWT=1                                        /* SYMBOLS 1 RASTER THICK
      PUTDEC=(AINT(FIRSTV).NE.FIRSTV).OR.(AINT(DELTAT).NE.DELTAT)
      NDEC=-1                                       /* DON'T PLOT DECIMAL PLACES IF
      IF(PUTDEC)NDEC=2                              /* THERE ARE NONE
      VAL=FIRSTV                                    /* CURRENT VALUE TO BE PLOTTED
C
C     PLOT 1 VALUE

105   CALL PLOT(X,Y,3)                              /* GET A POINT OF REFERENCE
      CALL RELPLT(XDISP,YDISP,3)                    /* NOW .2 IN FROM TIC MK (IN Y)

C
C     THIS NUMBER JOCKEYING WILL ALLOW US TO CENTER THE # BELOW THE TIC MARK

      HWIDE=(FLOAT(NCHRS-1)*.11143+.074285)/2.  /* HALF-WIDTH OF PLOTTED #
      CALL RELPLT(-HWIDE,0.,3)
      CALL WHERE(RX,RY,FDUMMY)                      /* GET ABSOLUTE PEN ADDRESS
      CALL NUMBER (RX,RY,.12,VAL,THETA,NDEC)    /* PLOT NUMBER
      XDISP=XDISP+DISMAJ                            /* UPDATE FOR NEXT TIC MK
```

```
C

      VAL=VAL+DELTAT
      IF(XDISP+X.LE.XLIMIT)GO TO 105          /* REPEAT 'TIL END OF AXIS

      NPLT=0                                  /* RESET NO-PLOT FLAG

C
C   PLOT AXIS TITLE IF ANY

      LNWT=LNWT1
      IF(NCH.EQ.0)RETURN
      ICH=IABS(NCH)                           /* # CHARACTERS IN TITLE
      CALL PLOT(X,Y,3)
      YDISP=-.75
      IF(NCH.GE.0)YDISP=.70
      HWIDE=(FLOAT(ICH-1)*.214284+.142857)/2
      XDISP=AXLEN/2-HWIDE
      CALL RELPLT(XDISP,YDISP,3)
      CALL WHERE(RX,RY,FDUMMY)
      LNWT=2
      CALL SYMBOL(RX,RY,.25,TITLE,IDUMMY,THETA,ICH) /* PLOT AXIS TITLE
      LNWT=LNWT1
      RETURN
      END


C   RCTNGL, DIPLOT
C   PLOT A RECTANGLE
C
      SUBROUTINE RCTNGL(X,Y,H,W,THETA,IZ)
      INTEGER*4  IZ
      REAL       X,Y,H,W,THETA
C
C     PARAMETERS :
C                X,Y      DEFINE THE LOWER LEFT CORNER OF THE RECTANGLE
C                H AND W  ARE THE HEIGHT AND WIDTH, RESPECTIVELY
C                THETA    IS THE ANGLE THE RECTANGLE MAKES WITH THE X-AXIS
C                IZ       DETERMINES WHETHER OR NOT A LINE IS DRAWN FROM THE
C                         ORIGIN TO (X,Y) - 2 TO DRAW, 3 OTHERWISE
C
$INSERT LGDEVICE.F77
C
      CALL PLOT(0.,0.,3)                      /* MOVE TO ORIGIN
      CALL PLOT(X,Y,IZ)                       /* THEN TO LOWER-LEFT CORNER
      TSIN=SIND(THETA)                        /* SET UP ROTATION VALUES
      TCOS=COSD(THETA)
C
      CALL RELPLT(W,0.,2)                     /* PLOT THE RECTANGLE
      CALL RELPLT(0.,H,2)
      CALL RELPLT(-W,0.,2)
      CALL RELPLT(0.,-H,2)
C
      RETURN
      END

C
C     SET   ANGLE
C
      SUBROUTINE ROTATE(THETA)
```

```
C

      REAL*4      THETA

C
C             THETA  THE ANGLE (IN DEGREES) FROM THE X-AXIS
C
$INSERT LGDEVICE.F77

      TCOS=COSD(THETA)
      TSIN=SIND(THETA)

      RETURN
      END


C
C     SET SCALING FACTOR
C
      SUBROUTINE FACTOR(FAC)

      REAL*4      FAC

C
C     PARAMETERS :
C                 FAC  IS THE SCALING FACTOR, IE:
C                 FAC=2.  DOUBLES THE SIZE OF THE PLOT

$INSERT LGDEVICE.F77

      INTEGER*2 LDEV

      IF(FAC.LE.0.)GO TO 5                    /* ILLEGAL FACTOR

C     RESET LAST-UPDATED  X & Y

      LASTX=LASTX*(FAC/SFACTR)
      LASTY=LASTY*(FAC/SFACTR)
C
      SFACTR=FAC                              /* ALL IS WELL
      RETURN

C     ILLEGAL SCALE FACTOR - COMPLAIN

5     CONTINUE
      WRITE(ERRDEV,10)FAC
10    FORMAT(/' BAD CALL TO FACTOR, FACTOR =',F10.3/)

      RETURN
      END

C
C     RETURN   COS(X), WHERE X IS IN DEGREES
C
      FUNCTION COSD(X)

      REAL*4    X

      COSD=COS(X/57.29578)
```

```
C


      RETURN
      END

C
C     SET CHARACTER SLANT
C
      SUBROUTINE CSLANT(SL)

      REAL*4      SL

C
C     PARAMETERS :
C                 SL   =  SLANT, IN DEGREES FROM THE +X AXIS
C

$INSERT LGDEVICE.F77

      SLANT=SIND(SL)                          /* GET SIN IN DEGREES

      RETURN
      END

C
C     RETURN SIN(X), WHERE X IS IN DEGREES
C
      FUNCTION SIND(Y)

      REAL*4    Y

      SIND=SIN(Y/57.29578)

      RETURN
      END


C
C     RETURN CURRENT PEN POSITION
C
      SUBROUTINE WHERE(X,Y,F)

      REAL*4      X,Y,F
C
C     PARAMETERS :
C                 X,Y   IS THE CURRENT CURSOR POSITION
C                 F     IS SCALE FACTOR
C                 NOTE: SCALING FACTOR IS TAKEN INTO CONSIDERATION

$INSERT LGDEVICE.F77

      X = LASTX
      Y = LASTY
      F = SFACTR

      RETURN
      END


C
C     PLOT DOTTED/DASHED VECTOR
```

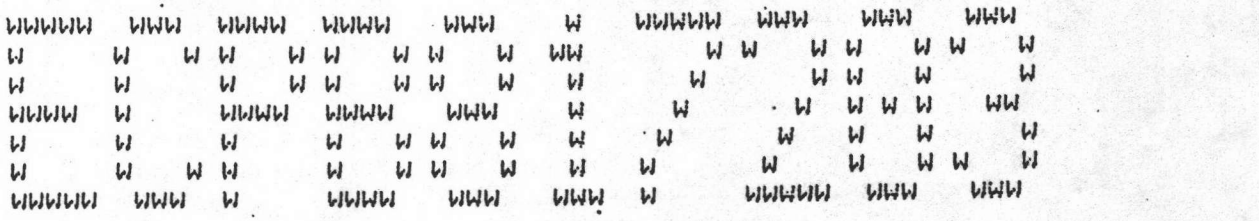
```
C
C
C
      SUBROUTINE DASHLN(X1,Y1,X2,Y2,MODE)

      INTEGER*4  MODE
      REAL*4     X1,Y1,X2,Y2

C
C      PARAMETERS :
C                 X1,Y1   =  START OF VECTOR
C                 X2,Y2   =  END OF VECTOR
C                 MODE    =  TYPE OF LINE TO BE DRAWN, AS FOLLOWS:
C                            0 - SOLID LINE
C                            1 - DOTTED LINE, 50 DOTS/INCH
C                            2 - DOTTED LINE, 25 DOTS/INCH
C                            3 - DOTTED LINE, 10 DOTS/INCH
C                            4 - DOTTED LINE, 5 DOTS/INCH
C                            5 - DASHED LINE, 1/16 INCH DASHES
C                            6 - DASHED LINE, 1/8 INCH DASHES
C                            7 - DASHED LINE, 1/4 INCH DASHES
C                            8 - DASHED LINE, 1/2 INCH DASHES
C                            9 - DASHED LINE, 1 INCH DASHES
C
      INTEGER*4 I,NPTS
      REAL*4    C,D,DELX,DELY,RLEN,S,SP,X,XD,XSP,Y,YD,YSP
C
c      write(15,*)'dashln =1',x1,y1,x2,y2,mode
      IF(MODE.EQ.0)GO TO 10                    /* PROCESS SOLID LINE
C
      DELX=X2-X1
      DELY=Y2-Y1
      RLEN=SQRT(ABS(DELX)**2.+ABS(DELY)**2.)          /* LINE LENGTH
      IF (RLEN.EQ.0.) RETURN
c      write(15,*)'dashln =2',MODE
      S=DELY/RLEN                              /* SIN(ANGLE)
      C=DELX/RLEN                              /* COS(ANGLE)
C
      CALL PLOT(X1,Y1,3)/* MOVE PEN TO STARTING POSITION
C
      GO TO (10,20,30,40,50,60,70,80,90,100),MODE+1
C
C
C   DRAW STRAIGHT LINE
10    CALL PLOT(X1,Y1,3)
      CALL PLOT(X2,Y2,2)
      RETURN
C
C   DOTTED, 50 DOTS/INCH
20    D=.02                                    /* (1/50)
22    NPTS=RLEN/D                              /* # POINTS TO DRAW
      X=X1                                     /* CURRENT X
      Y=Y1                                     /* CURRENT Y
C
      DO 28 I=1,NPTS
      CALL PLOT(X,Y,3)
      CALL PLOT(X,Y,2)                         /* (DRAW 1 POINT)
      X=X+C*D                                  /* UPDATE CURRENT X,Y
28    Y=Y+S*D
      RETURN
C
C   DOTTED, 25 DOTS/INCH
```

```
C
30      D=.04                                   /* (1/25)
        GO TO 22
C
C   DOTTED, 10 DOTS.INCH
40      D=.1                                    /* (1/10)
        GO TO 22
C
C   DOTTED, 5 DOTS/INCH
50      D=.2                                    /*-(1/5)
        GO TO 22
C
C   DASHED, 1/16 INCH DASHES
60      D=.0625                                 /* (1/16)
62      SP=D/2.                                 /* SPREAD BETWEEN DASHES
        XD=D*C                                  /* CALC RELATIVE X,Y OFFSETS
        YD=D*S
        XSP=SP*C
        YSP=SP*S
C
        IF(D*2.+SP.LE.RLEN)GO TO 66
        NPTS=1                                  /* ROOM FOR 1 DASH ONLY
C
C   PLOT DASHED LINES
64      DO 65 I=1,NPTS
        CALL RELPLT(XD,YD,2)
65      CALL RELPLT(XSP,YSP,3)
        RETURN
C
66      NPTS=(RLEN-D)/(D+SP)+1
        SP=(RLEN-D*NPTS)/(NPTS-1)               /* NEW SPAVE BETWEEN LINES
        XSP=SP*C
        YSP=SP*S
        GO TO 64
C
C   DASHED, 1/8 INCH DASHES
70      D=0.125                                 /* (1/8)
        GO TO 62
C
C   DASHED, 1/4 INCH DASHES
80      D=0.25                                  /* (1/4)
        GO TO 62
C
C   DASHED, 1/2 INCH DASHES
90      D=0.5
        GO TO 62
C
C   DASHED, 1 INCH DASHES
100     D=1.0
        GO TO 62
C
        END
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>CNTRLTMN. F77

****



****

Label: PRT009 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>CNTRLTMN. F77
File last modified: 87-08-27. 14:31:32. Thu

Spooled: 87-09-18. 10:37:28. Fri [Spooler rev 19.4]
Started: 87-09-18. 10:46:28. Fri on: PRO by: PRO

```
C
C
C
C     ENTER GRAPHIC MODE
C
      SUBROUTINE  ENTGRP

      EXTERNAL    TNDUA
      INTRINSIC   CHAR, INTS

      CHARACTER*1 ESC, DOL, SIX
      CHARACTER*3 ENGRPH

      ESC    = CHAR(27)
      DOL    = CHAR(36)
      SIX    = CHAR(54)

      ENGRPH = ESC//DOL//'E'
      CALL TNDUA( ENGRPH, INTS(3))
      ENGRPH = ESC//DOL//SIX              /* ENTER GRAPHIC MODE
      CALL TNDUA( ENGRPH, INTS(3))

      RETURN
      END

C
C     ENTER TEXT MODE
C
      SUBROUTINE ENTTXT

      EXTERNAL    TNDUA
      INTRINSIC   CHAR, INTS

      CHARACTER*1 ESC, DOL, QUES
      CHARACTER*3 ENTEXT

      ESC    = CHAR(27)
      DOL    = CHAR(36)
      QUES   = CHAR(63)

      ENTEXT = ESC//DOL//QUES             /* ENTER TEXT MODE
      CALL TNDUA( ENTEXT, INTS(3))
      ENTEXT = ESC//DOL//'P'
      CALL TNDUA( ENTEXT, INTS(3))

      RETURN
      END

C
C     ENTER NATIVE MODE
C
      SUBROUTINE  ENTNAT

      EXTERNAL    TNDUA
      INTRINSIC   CHAR, INTS

      CHARACTER*1 ESC, DOL
      CHARACTER*3 ENGRPH

      ESC    = CHAR(27)
      DOL    = CHAR(36)
      ENGRPH = ESC//DOL//'D'              /* ENTER NATIVE MODE
```

```
C

   10 CALL TNOUA( ENGRPH, INTS(3))

      RETURN
      END

C
C     ENTER TEKTRONIX MODE
C
      SUBROUTINE  ENTTEK

      EXTERNAL.     TNOUA
      INTRINSIC     CHAR, INTS

      CHARACTER*1 ESC, DOL
      CHARACTER*3 ENGRPH

      ESC    = CHAR(27)
      DOL.   = CHAR(36)
      ENGRPH = ESC//DOL//'1'                /* ENTER TEKTRONIX MODE

   10 CALL TNOUA( ENGRPH, INTS(3))

      RETURN
      END

C
C     CLEAR TEXT SCREEN
C
      SUBROUTINE CLRTXT

      EXTERNAL.     TNOUA
      INTRINSIC     CHAR, INTS

      CHARACTER     ESC, QUES
      CHARACTER*2 CLEARS

      ESC    = CHAR(27)
      QUES   = CHAR(63)
      CLEARS = ESC//QUES

      CALL TNOUA (CLEARS, INTS(2))

      RETURN
      END

C
C     CLEAR GRAPHIC TEKTRONIX
C
      SUBROUTINE CLRTEK

      EXTERNAL.     TNOUA
      INTRINSIC     CHAR, INTS

      CHARACTER*1 ESC, LCTRL
      CHARACTER*4 CLRSCR2

      ESC     = CHAR(27)
      LCTRL   = CHAR(12)
      CLRSCR2 = ESC//LCTRL
```

```
C

      CALL TNOUA (CLRSCR2, INTS(2))

      RETURN
      END

C
C     CLEAR GRAPHIC SCREEN0 & SCREEN1
C
      SUBROUTINE CLRGRP2

      EXTERNAL    TNOUA
      INTRINSIC   CHAR, INTS

      CHARACTER   ESC
      CHARACTER*4 CLRSCR2

      ESC     = CHAR(27)
      CLRSCR2 = ESC//'['//'2'//'x'

      CALL TNOUA (CLRSCR2, INTS(4))

      RETURN
      END

C
C     CLEAR GRAPHIC SCREEN0
C
      SUBROUTINE CLRGRP0

      EXTERNAL    TNOUA
      INTRINSIC   CHAR, INTS

      CHARACTER   ESC
      CHARACTER*4 CLRSCR0

      ESC     = CHAR(27)
      CLRSCR0 = ESC//'['//'0'//'x'

      CALL TNOUA (CLRSCR0, INTS(4))

      RETURN
      END

C
C     CLEAR GRAPHIC SCREEN1
C
      SUBROUTINE CLRGRP1

      EXTERNAL    TNOUA
      INTRINSIC   CHAR, INTS

      CHARACTER   ESC
      CHARACTER*4 CLRSCR1

      ESC     = CHAR(27)
      CLRSCR1 = ESC//'['//'1'//'x'

      CALL TNOUA (CLRSCR1, INTS(4))
```

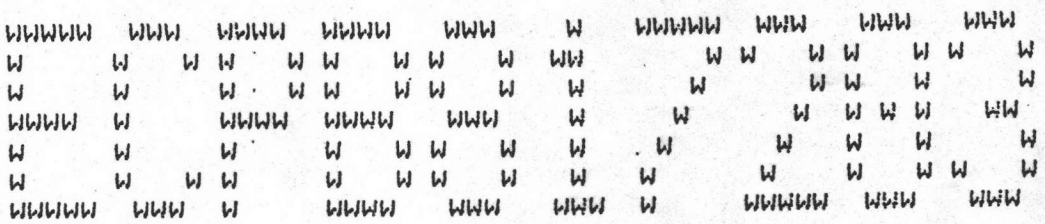
```
C


      RETURN
      END


C:
C     SELECT GRAPHIC SCREEN
C:
      SUBROUTINE  SELSCR(ISCR)

      EXTERNAL    TNDUA
      INTRINSIC   CHAR, INTS

      INTEGER*2   ISCR
      CHARACTER   ESC
      CHARACTER*3 SCR
      CHARACTER*4 CLRSCR1

      IF (ISCR.GT.3) ISCR=3
      IF (ISCR.LT.0) ISCR=0
      WRITE(SCR,25)ISCR
25    FORMAT(I3)
      ESC      = CHAR(27)
      CLRSCR1 = ESC//'['//SCR(3:3)//'w'

      CALL TNDUA (CLRSCR1, INTS(4))

      RETURN
      END

C:
C     ERASE POINT
C
      SUBROUTINE  ERAPOINT (IX, IY)

      EXTERNAL    TNDUA, ATOI, ITOA, T1IN, DELBLANK, LENGCHAR
      INTRINSIC   INTS, OR, AND, NOT, LS, RS, ICHAR, CHAR

      INTEGER*2   ITEMP1, ITEMP2, ITEMP3, ITEMP4, LOOP1, LOOP2
      INTEGER*2   IX, IY, IROW, IXBYTE, IXBIT, LENGCHAR
      CHARACTER*1 CTEMP1, CTEMP2, CTEMP3, CTEMP4, ESC
      CHARACTER*3 BROW, ROW, BCOL, COL
      CHARACTER*120 CTRLGRAP, SCRNBUFF
      ESC = CHAR(27)

C     RECEIVE DATA IX, IY FOR PLOTTING & CONVERT TO OPERATIONS
      IROW = 299 - IY
      WRITE(BROW, 10)IROW
10    FORMAT(I3)
      CALL DELBLANK(BROW, ROW)
      ITEMP1 = LENGCHAR(ROW)
      CTRLGRAP = ESC//'C'//ROW(1:ITEMP1)//';1;1#'
      ITEMP1 = LENGCHAR(CTRLGRAP)

      CALL TNDUA (CTRLGRAP, INTS(ITEMP1))
      DO 20 LOOP1 = 1,60
        CALL T1IN (ITEMP1)
        CALL ITOA (ITEMP1, CTEMP1, CTEMP2)
        ITEMP2 = LOOP1*2
        ITEMP3 = ITEMP2+1
        SCRNBUFF(ITEMP2:ITEMP2) = CTEMP1
```

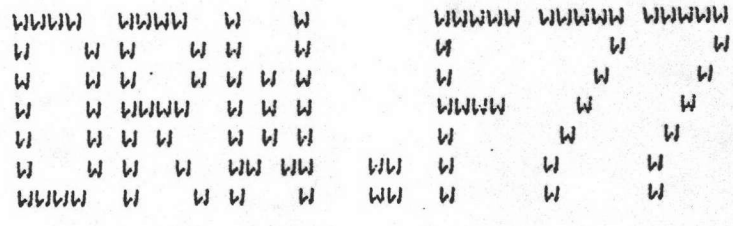
```
C
          SCRNBUFF(ITEMP3:ITEMP3) = CTEMP2
20      CONTINUE
C
C       MARK POINT
C
        ITEMP1 = IX/6
        IXBYTE = ITEMP1+1
        ITEMP2 = IX-ITEMP1
        IXBIT  = 5-ITEMP2
        CTEMP1 = SCRNBUFF(IXBYTE:IXBYTE)
        ITEMP1 = ICHAR(CTEMP1)
        ITEMP2 = LS(:000001,IXBIT)
        ITEMP2 = NOT (ITEMP2)
        ITEMP1 = AND(ITEMP1,ITEMP2)
        CALL ITOA (ITEMP1,CTEMP1,CTEMP2)  /* CTEMP2 IS MARKED BYTE
C
C       SEND DATA FOR PLOTTING AT THAT POINT
C
        ITEMP1 = (IXBYTE-1)*6
        WRITE(BCOL,10)ITEMP1
        CALL DELBLANK (BCOL,COL)
        ITEMP1 = LENGCHAR(ROW)
        ITEMP2 = LENGCHAR(COL)
        CTRLGRAP = ESC//'7'//COL(1:ITEMP1)//';'
       1//ROW(1:ITEMP1)//';1'//CTEMP2//'*'
        ITEMP1 = LENGCHAR(CTRLGRAP)
        CALL TNOUA (CTRLGRAP,INTS(ITEMP1))

        RETURN
        END

        SUBROUTINE  ITOA (ITEMP1,CTEMP1,CTEMP2)

        INTRINSIC   CHAR,RS

        INTEGER*2   ITEMP1,ITEMP2
        CHARACTER*1 CTEMP1,CTEMP2

        ITEMP2 = ITEMP1
        CTEMP2 = CHAR(ITEMP2)
        ITEMP2 = RS(ITEMP2,8)
        CTEMP1 = CHAR(ITEMP2)

        RETURN
        END

        SUBROUTINE  ATOI (ITEMP1,CTEMP1,CTEMP2)

        INTRINSIC   ICHAR,LS,AND,OR

        INTEGER*2   ITEMP1,ITEMP2,ITEMP3
        CHARACTER*1 CTEMP1,CTEMP2

        ITEMP1 = ICHAR(CTEMP1)
        ITEMP1 = LS(ITEMP1,8)
        ITEMP2 = ICHAR(CTEMP2)
        ITEMP1 = AND(ITEMP1,:177400)
        ITEMP2 = AND(ITEMP2,:000377)
        ITEMP1 = OR(ITEMP1,ITEMP2)
```

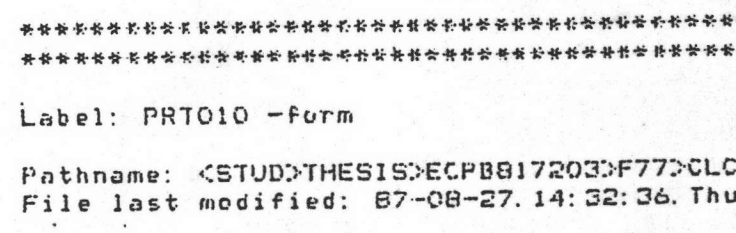
```
C
      RETURN
      END

C
C     FIND LENGTH OF CHARACTER
C
      INTEGER*2 FUNCTION LENGCHAR (CHARARGU)
      INTRINSIC         LEN, ICHAR
      INTEGER*2         I, J, K, L
      CHARACTER*(*)     CHARARGU
      CHARACTER*1       TESTCHAR

      J=LEN(CHARARGU)
      L=J
      DO 1000 I=1, J
        K=J-I+1
        TESTCHAR=CHARARGU(K:K)
        IF (ICHAR(TESTCHAR) .NE. 160) GOTO 10
        L=L-1
1000  CONTINUE
10    CONTINUE

      LENGCHAR=L

      RETURN
      END

C
C     DELETE BLANK
C
      SUBROUTINE DELBLANK(DCHAR, OCHAR)

      INTRINSIC      LEN, ICHAR

      CHARACTER*(*) DCHAR, OCHAR
      INTEGER*2      I, J, K1, K2


      J = LEN(DCHAR)
      DO 1000 I = 1, J
        IF (ICHAR(DCHAR(I:I)) .NE. 160) GOTO 100
1000  CONTINUE

      GOTO 200

100   CONTINUE
      K2 = 1
      OCHAR = ''
      DO 2000 K1 = 1, J
        OCHAR(K2:K2) = DCHAR(K1:K1)
        K2 = K2+1
2000  CONTINUE

200   CONTINUE

      RETURN
      END
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>DRW.F77







Label: PRT010 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>DRW.F77
File last modified: 87-08-27.14:32:36.Thu

Spooled: 87-09-18.10:37:28.Fri [Spooler rev 19.4]
Started: 87-09-18.10:47:44.Fri on: PRO by: PRO

```
C
C
C
C       PRODUCE LINE
C
        SUBROUTINE BDRWLINE(IX1B, IY1B, IX2B, IY2B)

        INTEGER*2   IX1A, IY1A, IX2A, IY2A
        INTEGER*2   IX1B, IY1B, IX2B, IY2B
        INTEGER*4   IX1, IY1, IX2, IY2
        INTEGER*4   IX, IY, IYCHANGE
        INTEGER*4   IDX, IDY, IP, ICNT1, ICNT2, ICASE
        REAL*4      DT, TEMP1, TEMP2, TEMP3, TEMP4

c        write(15,*)'bdrw para', ix1b, iy1b, ix2b, iy2b
c        call entnat
        TEMP1 = (REAL(IX1B)*719.)/1023.
        IX1A  = INTS(TEMP1)
        TEMP2 = (REAL(IY1B)*291.)/779.  +8.
        IY1A  = INTS(TEMP2)
        TEMP3 = (REAL(IX2B)*719.)/1023.
        IX2A  = INTS(TEMP3)
        TEMP4 = (REAL(IY2B)*291.)/779.  +8.
        IY2A  = INTS(TEMP4)


        IX1 = INTL(IX1A)
        IY1 = INTL(IY1A)
        IX2 = INTL(IX2A)
        IY2 = INTL(IY2A)
c        write(15,*)'bdrw IXY', ix1, iy1, ix2, iy2

        IF ((IX1.EQ.IX2).AND.(IY1.EQ.IY2)) CALL FDRWPOINT(IX1A, IY1A)
c        write(15,*)'bdrw CASE POINT'
        IF ((IX1.EQ.IX2).AND.(IY1.EQ.IY2)) GOTO 9999

c       write(15,*)'bdrw CASE IX1=IX2'
        IF (IX1.EQ.IX2) GOTO 2000
c        write(15,*)'bdrw CASE IY1=IY2'
        IF (IY1.EQ.IY2) GOTO 3000

        DT = REAL(IY1-IY2)/REAL(IX1-IX2)
        IYCHANGE = 1
c        write(15,*)'bdrw idt=', DT, IFIX(DT)
c        write(15,*)'bdrw CASE DT=1.'
        IF (DT.EQ.1.)  GOTO 5000
        IF (DT.EQ.-1.) IYCHANGE = -1
c        write(15,*)'bdrw CASE DT=-1.'
        IF (DT.EQ.-1.) GOTO 5000

        IF ((DT.LT.1.).AND.(DT.GT.0.)) ICASE = 1
        IF (DT.GT.1.)                  ICASE = 2
        IF (DT.LT.-1.)                 ICASE = 3
        IF ((DT.GT.-1.).AND.(DT.LT.0.)) ICASE = 4

        IF ((ICASE.NE.2).AND.(ICASE.NE.3)) GOTO 100
          IX1 = INTL(IY1A)
          IX2 = INTL(IY2A)
          IY1 = INTL(IX1A)
          IY2 = INTL(IX2A)

  100   CONTINUE
```

```
C
      IDX   = IABS(IX1-IX2)
      IDY   = IABS(IY1-IY2)
      IP    = 2*IDY-IDX
      ICT1 = 2*IDY
      ICT2 = 2*(IDY-IDX)

C      write(15,*)'bdrw icase=',icase
      IF (IX1.LE.IX2) GOTO 200
        IX = IX2
        IY = IY2
        IXEND = IX1
        GOTO 500

200   CONTINUE
        IX = IX1
        IY = IY1
        IXEND = IX2
        GOTO 500

500   CONTINUE
      IF ((ICASE.EQ.1).OR.(ICASE.EQ.4))
     X    CALL FDRWPOINT(INTS(IX),INTS(IY))
      IF ((ICASE.EQ.2).OR.(ICASE.EQ.3))
     X    CALL FDRWPOINT(INTS(IY),INTS(IX))

      IYCHANGE = 1
      IF ((ICASE.EQ.3).OR.(ICASE.EQ.4)) IYCHANGE = -1

1000  CONTINUE
      IF (IX.GE.IXEND) GOTO 9999
        IX = IX + 1
        IF (IP.GE.0) GOTO 600
          IP = IP + ICT1
          GOTO 700

600     CONTINUE
          IP = IP + ICT2
          IY = IY +IYCHANGE
          GOTO 700

700   CONTINUE
      IF ((ICASE.EQ.1).OR.(ICASE.EQ.4))
     X    CALL FDRWPOINT(INTS(IX),INTS(IY))
      IF ((ICASE.EQ.2).OR.(ICASE.EQ.3))
     X    CALL FDRWPOINT(INTS(IY),INTS(IX))
      GOTO 1000

2000  CONTINUE
      IF (IY1.LE.IY2) GOTO 2100
        IY = IY2
        IYEND = IY1
        GOTO 2200

2100  CONTINUE
        IY = IY1
        IYEND = IY2
        GOTO 2200

2200  CONTINUE
      DO 2300 J = IY,IYEND
```

```
C
        CALL FDRWPOINT(INTS(IX1),INTS(J))
2300    CONTINUE
        GOTO 9999

3000    CONTINUE
        IF (IX1.LE.IX2) GOTO 3100
          IX = IX2
          IXEND = IX1
          GOTO 3200

3100    CONTINUE
          IX = IX1
          IXEND = IX2
          GOTO 3200

3200    CONTINUE
        DO 3300 J = IX,IXEND
          CALL FDRWPOINT(INTS(J),INTS(IY1))
3300    CONTINUE
        GOTO 9999

5000    CONTINUE
        IF (IX1.LE.IX2) GOTO 5100
          IX    = IX2
          IY    = IY2
          IXEND = IX1
          GOTO 5200

5100    CONTINUE
          IX    = IX1
          IY    = IY1
          IXEND = IX2
          GOTO 5200

5200    CONTINUE
        DO 5300 J = IX,IXEND
          CALL FDRWPOINT(INTS(J),INTS(IY))
          IY = IY + IYCHANGE
5300    CONTINUE
        GOTO 9999

9999    CONTINUE
c        call enttek
        RETURN
        END

C
C       PRODUCE LINE
C
        SUBROUTINE TBDRWLINE(IX1B,IY1B,IX2B,IY2B)

        INTEGER*2   IX1A,IY1A,IX2A,IY2A
        INTEGER*2   IX1B,IY1B,IX2B,IY2B
        INTEGER*4   IX1,IY1,IX2,IY2
        INTEGER*4   IX,IY,IYCHANGE
        INTEGER*4   IDX,IDY,IP,ICNT1,ICNT2,ICASE
        REAL*4      DT,TEMP1,TEMP2,TEMP3,TEMP4

c        write(15,*)'bdrw para',ix1b,iy1b,ix2b,iy2b
        call entnat
```

```
C

      TEMP1 = (REAL(IX1B)*719.)/1023.
      IX1A  = INTS(TEMP1)
      TEMP2 = (REAL(IY1B)*291.)/779. +8.
      IY1A  = INTS(TEMP2)
      TEMP3 = (REAL(IX2B)*719.)/1023.
      IX2A  = INTS(TEMP3)
      TEMP4 = (REAL(IY2B)*291.)/779. +8.
      IY2A  = INTS(TEMP4)


      IX1 = INTL(IX1A)
      IY1 = INTL(IY1A)
      IX2 = INTL(IX2A)
      IY2 = INTL(IY2A)
C      write(15,*)'bdrw IXY',ix1,iy1,ix2,iy2

      IF ((IX1.EQ.IX2).AND.(IY1.EQ.IY2)) CALL BDRWPOINT(IX1A,IY1A)
C      write(15,*)'bdrw CASE POINT'
      IF ((IX1.EQ.IX2).AND.(IY1.EQ.IY2)) GOTO 9999

C      write(15,*)'bdrw CASE IX1=IX2'
      IF (IX1.EQ.IX2) GOTO 2000
C      write(15,*)'bdrw CASE IY1=IY2'
      IF (IY1.EQ.IY2) GOTO 3000

      DT = REAL(IY1-IY2)/REAL(IX1-IX2)
      IYCHANGE = 1
C      write(15,*)'bdrw idt=',DT,IFIX(DT)
C      write(15,*)'bdrw CASE DT=1.'
      IF (DT.EQ.1.)  GOTO 5000
      IF (DT.EQ.-1.) IYCHANGE = -1
C      write(15,*)'bdrw CASE DT=-1.'
      IF (DT.EQ.-1.) GOTO 5000

      IF ((DT.LT.1.).AND.(DT.GT.0.)) ICASE = 1
      IF (DT.GT.1.)                  ICASE = 2
      IF (DT.LT.-1.)                 ICASE = 3
      IF ((DT.GT.-1.).AND.(DT.LT.0.)) ICASE = 4

      IF ((ICASE.NE.2).AND.(ICASE.NE.3)) GOTO 100
        IX1 = INTL(IY1A)
        IX2 = INTL(IY2A)
        IY1 = INTL(IX1A)
        IY2 = INTL(IX2A)

100   CONTINUE
      IDX   = IABS(IX1-IX2)
      IDY   = IABS(IY1-IY2)
      IP    = 2*IDY-IDX
      ICT1 = 2*IDY
      ICT2 = 2*(IDY-IDX)

C      write(15,*)'bdrw icase=',icase
      IF (IX1.LE.IX2) GOTO 200
        IX = IX2
        IY = IY2
        IXEND = IX1
        GOTO 500

200   CONTINUE
```

```
C
        IX = IX1
        IY = IY1
        IXEND = IX2
        GOTO 500

500    CONTINUE
       IF ((ICASE.EQ.1).OR.(ICASE.EQ.4))
      X    CALL BDRWPOINT(INTS(IX),INTS(IY))
       IF ((ICASE.EQ.2).OR.(ICASE.EQ.3))
      X    CALL BDRWPOINT(INTS(IY),INTS(IX))

       IYCHANGE = 1
       IF ((ICASE.EQ.3).OR.(ICASE.EQ.4)) IYCHANGE = -1

1000   CONTINUE
       IF (IX.GE.IXEND) GOTO 9999
         IX = IX + 1
         IF (IP.GE.0) GOTO 600
           IP = IP + ICT1
           GOTO 700

600      CONTINUE
           IP = IP + ICT2
           IY = IY +IYCHANGE
           GOTO 700

700    CONTINUE
       IF ((ICASE.EQ.1).OR.(ICASE.EQ.4))
      X    CALL BDRWPOINT(INTS(IX),INTS(IY))
       IF ((ICASE.EQ.2).OR.(ICASE.EQ.3))
      X    CALL BDRWPOINT(INTS(IY),INTS(IX))
       GOTO 1000

2000   CONTINUE
       IF (IY1.LE.IY2) GOTO 2100
         IY = IY2
         IYEND = IY1
         GOTO 2200

2100   CONTINUE
         IY = IY1
         IYEND = IY2
         GOTO 2200

2200   CONTINUE
       DO 2300 J = IY,IYEND
         CALL BDRWPOINT(INTS(IX1),INTS(J))
2300   CONTINUE
       GOTO 9999

3000   CONTINUE
       IF (IX1.LE.IX2) GOTO 3100
         IX = IX2
         IXEND = IX1
         GOTO 3200

3100   CONTINUE
         IX = IX1
         IXEND = IX2
         GOTO 3200
```

```
C

3200  CONTINUE
      DO 3300 J = IX, IXEND
        CALL BDRWPOINT(INTS(J), INTS(IY1))
3300  CONTINUE
      GOTO 9999

5000  CONTINUE
      IF (IX1.LE.IX2) GOTO 5100
        IX    = IX2
        IY    = IY2
        IXEND = IX1
        GOTO 5200

5100  CONTINUE
        IX    = IX1
        IY    = IY1
        IXEND = IX2
        GOTO 5200

5200  CONTINUE
      DO 5300 J = IX, IXEND
        CALL BDRWPOINT(INTS(J), INTS(IY))
        IY = IY + IYCHANGE
5300  CONTINUE
      GOTO 9999

9999  CONTINUE
      call enttek
      RETURN
      END


C
C     DRAW POINT
C
      SUBROUTINE  BDRWPOINT (IX, IY)

$INSERT IMGPAPER.F77

      EQUIVALENCE ( INFRCH, CHFRIN )

      EXTERNAL    TNOUA, DELBLANK
      EXTERNAL    ITOA, LENGCHAR
      INTRINSIC   INTS, OR, AND, NOT, LS, RS, ICHAR, CHAR

      INTEGER*2   ITEMP1, ITEMP2, ITEMP3, ITEMP4, LOOP1, LOOP2
      INTEGER*2   IX, IY, IROW, IXBYTE, IXBIT, LENGCHAR, INFRCH
      CHARACTER*1 CTEMP1, CTEMP2, CTEMP3, CTEMP4, ESC
      CHARACTER*2 CHFRIN
      CHARACTER*3 BROW, ROW, BCOL, COL
      CHARACTER*120 CTRLGRAP, SCRNBUFF

      ESC = CHAR(:233)

      IF (IX.LT.0)    IX=0
      IF (IY.LT.0)    IY=0
      IF (IX.GT.719)  IX=719
      IF (IY.GT.299)  IY=299
C
```

```
C
C       RECEIVE DATA IX, IY FOR PLOTTING & CONVERT TO OPERATIONS
C
        IROW = 300 - IY

C
C       MARK POINT
C
        ITEMP1 = IX/6
        IXBYTE = ITEMP1+1
        ITEMP2 = IX-(ITEMP1*6)
        IXBIT  = 5-ITEMP2
        ITEMP2 = LS(INTS(:000001),IXBIT)
        CTEMP1 = PAPER(IROW)(IXBYTE:IXBYTE)
        ITEMP1 = ICHAR(CTEMP1)
        INFRCH = OR(ITEMP1,ITEMP2)
        INFRCH = AND(INFRCH,INTS(:177577))
        INFRCH = OR(INFRCH,INTS(:000100))     /* CHFRIN(2:2) IS MARKED BYTE
        PAPER(IROW)(IXBYTE:IXBYTE) = CHFRIN(2:2)

C
C       SEND DATA FOR PLOTTING AT THAT POINT

        ITEMP1 = (IXBYTE-1)*6
        WRITE(BCOL,10)ITEMP1
10      FORMAT(I3)
        CALL DELBLANK (BCOL,COL)
        ITEMP1 = LENGCHAR(COL)
        ITEMP3 = IROW-1
        WRITE(BROW,10)ITEMP3
        CALL DELBLANK(BROW,ROW)
        ITEMP2 = LENGCHAR(ROW)

        CTRLGRAP = ESC//'7'//COL(1:ITEMP1)//';'
       1//ROW(1:ITEMP2)//';1'//PAPER(IROW)(IXBYTE:IXBYTE)//'*'
        ITEMP1 = LENGCHAR(CTRLGRAP)

        CALL TNDUA (CTRLGRAP,INTS(ITEMP1))


        RETURN
        END

C
C       DRAW LINE BY NATIVE MODE
C
        SUBROUTINE  FDRWLINE(IX1,IY1,IX2,IY2)

        EXTERNAL  DRWPOINT
        INTRINSIC DSQRT


        INTEGER*2    IX,IY,IX1,IY1,IX2,IY2,LOOP,ENDLOOP
        REAL*8       TEMP1,TEMP2,TEMP3,TEMP4,TEMP5
        REAL*8       TEMP6,TEMP7,TEMP8,TEMP9,TEMP10,UX,UY

c       write(15,*)'fdrwline 01',ix1,iy1,ix2,iy2
        TEMP1 = (REAL(IX1)*719.)/1023.
        IX1A  = INTS(TEMP1)
        TEMP2 = (REAL(IY1)*291.)/779.  +8.
```

```
C

      IY1A   = INTS(TEMP2)
      TEMP3 = (REAL(IX2)*719.)/1023.
      IX2A   = INTS(TEMP3)
      TEMP4 = (REAL(IY2)*291.)/779.  +8.
      IY2A   = INTS(TEMP4)
c      write(15,*)'fdrwline 02',ix1a,iy1a,ix2a,iy2a

      TEMP5 =  DSQRT( ((TEMP3-TEMP1)**2)+((TEMP4-TEMP2)**2) )
      IF (TEMP5.EQ.0) GOTO 9999
      UX    =  (TEMP3-TEMP1)/TEMP5
      UY    =  (TEMP4-TEMP2)/TEMP5

      ENDLOOP  =  TEMP5+0.5

c      call entnat

      CALL FDRWPOINT(IX1A,IY1A)
      TEMP1 = IX1A
      TEMP2 = IY1A
      DO 20 LOOP = 1,ENDLOOP

         TEMP1 = TEMP1+UX
         IX    = TEMP1
         TEMP2 = TEMP2+UY
         IY    = TEMP2


         IF (UX.LT.0.) GOTO 100
         IF (IX.GT.IX2A)  IX = IX2A
         GOTO 1000
100      IF (IX.LT.IX2A)  IX = IX2A
         GOTO 1000

1000     CONTINUE
         IF (UY.LT.0.) GOTO 200
         IF (IY.GT.IY2A)  IY = IY2A
         GOTO 2000
200      IF (IY.LT.IY2A)  IY = IY2A
         GOTO 2000

2000     CALL FDRWPOINT(IX,IY)
c         write(15,*)'fdrline',ix,iy
20    CONTINUE
      CALL FDRWPOINT(IX2A,IY2A)

c      call enttek


9999  CONTINUE
      RETURN
      END

C
C     DRAW POINT
C
      SUBROUTINE  FDRWPOINT (IX,IY)

*INSERT INSPAPER.F77

      EQUIVALENCE ( INFRCH,CHFRIN )
```

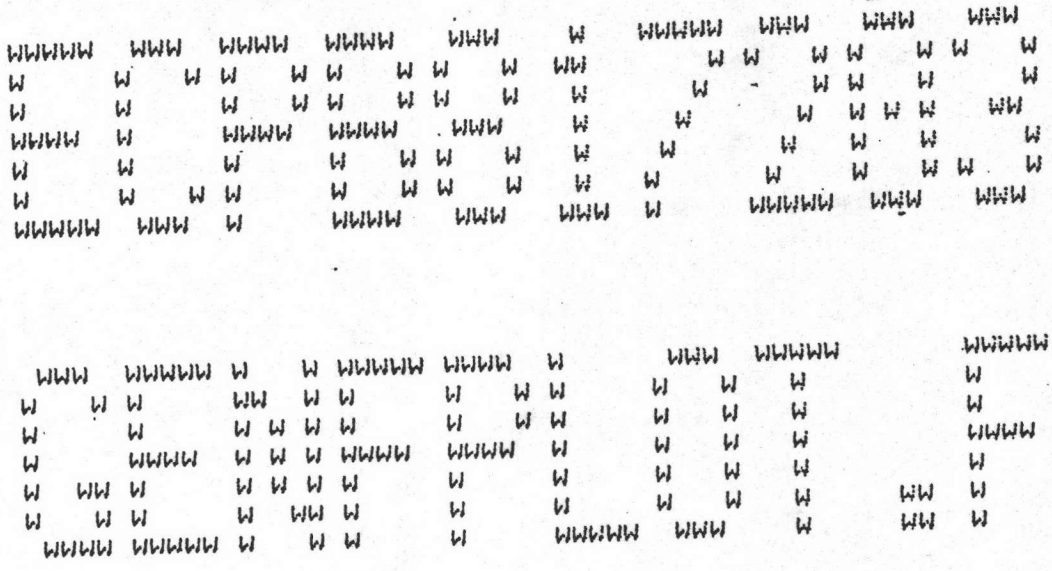
```
C
      EXTERNAL     TNOUA, DELBLANK
      EXTERNAL     ITOA, LENGCHAR
      INTRINSIC    INTS, OR, AND, NOT, LS, RS, ICHAR, CHAR

      INTEGER*2    ITEMP1, ITEMP2, ITEMP3, ITEMP4, LOOP1, LOOP2
      INTEGER*2    IX, IY, IROW, IXBYTE, IXBIT, LENGCHAR, INFRCH

      CHARACTER*1  CTEMP1, CTEMP2, CTEMP3, CTEMP4, ESC
      CHARACTER*2  CHFRIN
      CHARACTER*3  BROW, ROW, BCOL, COL
      CHARACTER*120 CTRLGRAP, SCRNBUFF


      ESC = CHAR(:233)

      IF (IX.LT.0)    IX=0
      IF (IY.LT.0)    IY=0
      IF (IX.GT.719)  IX=719
      IF (IY.GT.599)  IY=599
C
C     RECEIVE DATA IX, IY FOR PLOTTING & CONVERT TO OPERATIONS
C
      IROW = 600 - IY


C
C     MARK POINT
C
      ITEMP1 = IX/6
      IXBYTE = ITEMP1+1
      ITEMP2 = IX-(ITEMP1*6)
      IXBIT  = 5-ITEMP2
      ITEMP2 = LS(INTS(:000001), IXBIT)
      CTEMP1 = PAPER(IROW)(IXBYTE:IXBYTE)
      ITEMP1 = ICHAR(CTEMP1)
      INFRCH = OR(ITEMP1, ITEMP2)
      INFRCH = AND(INFRCH, INTS(:177577))
      INFRCH = OR(INFRCH, INTS(:000100))     /* CHFRIN(2:2) IS MARKED BYTE
      PAPER(IROW)(IXBYTE:IXBYTE) = CHFRIN(2:2)

      RETURN
      END

C
C     DRAW LINE BY NATIVE MODE
C
      SUBROUTINE  DRWLINE(IX1, IY1, IX2, IY2)

      EXTERNAL  DRWPOINT
      INTRINSIC DSQRT


      INTEGER*2    IX, IY, IX1, IY1, IX2, IY2, LOOP, ENDLOOP
      REAL*8       TEMP1, TEMP2, TEMP3, TEMP4, TEMP5
      REAL*8       TEMP6, TEMP7, TEMP8, TEMP9, TEMP10, UX, UY

C      write(15,*)'fdrwline 01', ix1, iy1, ix2, iy2
      TEMP1 = (REAL(IX1)*719.)/1023.
      IX1A  = INTS(TEMP1)
```

```
C

      TEMP2 = (REAL(IY1)*291.)/779.  +8.
      IY1A  = INTS(TEMP2)
      TEMP3 = (REAL(IX2)*719.)/1023.
      IX2A  = INTS(TEMP3)
      TEMP4 = (REAL(IY2)*291.)/779.  +8.
      IY2A  = INTS(TEMP4)
C      write(15,*)'fdrwline 02',ix1a,iy1a,ix2a,iy2a

      TEMP5 =  DSQRT( ((TEMP3-TEMP1)**2)+((TEMP4-TEMP2)**2) )
      IF (TEMP5.EQ.0) GOTO 9999
      UX    =  (TEMP3-TEMP1)/TEMP5
      UY    =  (TEMP4-TEMP2)/TEMP5

      ENDLOOP  =  TEMP5+0.5

      call entnat

      CALL DRWPOINT(IX1A,IY1A)
      TEMP1 = IX1A
      TEMP2 = IY1A
      DO 20 LOOP = 1,ENDLOOP

         TEMP1 = TEMP1+UX
         IX    = TEMP1
         TEMP2 = TEMP2+UY
         IY    = TEMP2


         IF (UX.LT.0.) GOTO 100
         IF (IX.GT.IX2A)  IX = IX2A
         GOTO 1000
100      IF (IX.LT.IX2A)  IX = IX2A
         GOTO 1000

1000     CONTINUE
         IF (UY.LT.0.) GOTO 200
         IF (IY.GT.IY2A)  IY = IY2A
         GOTO 2000
200      IF (IY.LT.IY2A)  IY = IY2A
         GOTO 2000

2000     CALL DRWPOINT(IX,IY)
C         write(15,*)'fdrline',ix,iy
20    CONTINUE
      CALL DRWPOINT(IX2A,IY2A)

      call enttek


9999  CONTINUE
      RETURN
      END

C
C     DRAW POINT
C
      SUBROUTINE  DRWPOINT (IX,IY)

*INSERT INGPAPER.F77
```

```
C

      EQUIVALENCE ( INFRCH,CHFRIN )

      EXTERNAL     TNDUA, DELBLANK
      EXTERNAL     ITOA, LENGCHAR
      INTRINSIC    INTS, OR, AND, NOT, LS, RS, ICHAR, CHAR

      INTEGER*2    ITEMP1, ITEMP2, ITEMP3, ITEMP4, LOOP1, LOOP2
      INTEGER*2    IX, IY, IROW, IXBYTE, IXBIT, LENGCHAR, INFRCH
      CHARACTER*1  CTEMP1, CTEMP2, CTEMP3, CTEMP4, ESC
      CHARACTER*2  CHFRIN
      CHARACTER*3  BROW, ROW, BCOL, COL
      CHARACTER*120 CTRLGRAP, SCRNBUFF

      ESC = CHAR(:233)

      IF (IX.LT.0)    IX=0
      IF (IY.LT.0)    IY=0
      IF (IX.GT.719) IX=719
      IF (IY.GT.599) IY=599
C
C     RECEIVE DATA IX,IY FOR PLOTTING & CONVERT TO OPERATIONS
C
      IROW = 300 - IY


C:
C     MARK POINT
C
      ITEMP1 = IX/6
      IXBYTE = ITEMP1+1
      ITEMP2 = IX-(ITEMP1*6)
      IXBIT  = 5-ITEMP2
      ITEMP2 = LS(INTS(:000001),IXBIT)
      CTEMP1 = PAPER(IROW)(IXBYTE:IXBYTE)
      ITEMP1 = ICHAR(CTEMP1)
      INFRCH = OR(ITEMP1,ITEMP2)
      INFRCH = AND(INFRCH,INTS(:177577))
      INFRCH = OR(INFRCH,INTS(:000100))     /* CHFRIN(2:2) IS MARKED BYTE
      PAPER(IROW)(IXBYTE:IXBYTE) = CHFRIN(2:2)

C:
C     SEND DATA FOR PLOTTING AT THAT POINT

      ITEMP1 = (IXBYTE-1)*6
      WRITE(BCOL,10)ITEMP1
10    FORMAT(I3)
      CALL DELBLANK (BCOL,COL)
      ITEMP1 = LENGCHAR(COL)
      ITEMP3 = IROW-1
      WRITE(BROW,10)ITEMP3
      CALL DELBLANK(BROW,ROW)
      ITEMP2 = LENGCHAR(ROW)

      CTRLGRAP = ESC//'7'//COL(1:ITEMP1)//';'
     1//ROW(1:ITEMP2)//';1'//PAPER(IROW)(IXBYTE:IXBYTE)//'*'
      ITEMP1 = LENGCHAR(CTRLGRAP)

      CALL TNDUA (CTRLGRAP,INTS(ITEMP1))
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>GENFPLOT.F77



Label: PRT011 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>GENFPLOT.F77
File last modified: 87-08-31.17:38:20.Mon

Spooled: 87-09-18.10:37:32.Fri [Spooler rev 19.4]
Started: 87-09-18.10:50:00.Fri on: PRO by: PRO

```
C
C
C
C       GENERATE VECTOR ON FILE
C
        SUBROUTINE GENFPLOT

        INTEGER*2   LENGCHAR, LENGLINE, LFPLTNAM, LFVECNAM
        INTEGER*2   IX, IY, IXP, IYP
        INTEGER*2   HEADRPIX, HEADRPIY, HEADXMAX, HEADYMAX
        INTEGER*2   HEADXHI , HEADYHI
        INTEGER*4   HEADLENG, HEADRESV
        INTEGER*4   IPEN
        CHARACTER*30 FPLTNAME, LINEBUFF, FVECNAME
        CHARACTER*1 ADUMMY

        INTEGER*2       ITEMP1, ITEMP2, ITEMP3, ITEMP4, ITEMP5, ITEMP6
        INTEGER*2       MARK, TESTNK, INFRCH
        INTEGER*2       IHEADFOR, IGCONTRO
        CHARACTER*4     FILEBUFF
        CHARACTER*2     HEADFORM, GCONTROL, CHFRIN
        CHARACTER*120 LPAPER, LPAPER2
        EQUIVALENCE     (IHEADFOR, HEADFORM), (IGCONTRO, GCONTROL)
        EQUIVALENCE     (INFRCH, CHFRIN)

$INSERT IMGPAPER.F77

C
C       CLEAR PAPER
        DO 1100 J1 = 1, 600
          DO 1200 J2 = 1, 120
            PAPER(J1)(J2:J2) = CHAR(:140300)
1200      CONTINUE
1100    CONTINUE


C
C       OPEN VECTOR FILE

        LINEBUFF = 'ENTER VECTOR FILENAME : '
        LENGLINE = LENGCHAR(LINEBUFF)+1
        CALL TNOUA(LINEBUFF, LENGLINE)
        READ(1, 17)FPLTNAME
17      FORMAT(A30)
        LFPLTNAM = LENGCHAR(FPLTNAME)
        IF (LFPLTNAM.EQ.0) FPLTNAME = 'VECTOR'
        LFPLTNAM = LENGCHAR(FPLTNAME)

        FVECNAME = FPLTNAME
        LFVECNAM = LFPLTNAM

c        write(15,*)'genfplot 01', fVECname(1:lfVECnam)
        OPEN(77, FILE=FVECNAME(1:LFVECNAM))

C
C       OPEN PLOTTED FILE

        LINEBUFF = 'ENTER PLOTTED FILENAME : '
        LENGLINE = LENGCHAR(LINEBUFF)+1
        CALL TNOUA(LINEBUFF, LENGLINE)
        READ(1, 17)FPLTNAME
        LFPLTNAM = LENGCHAR(FPLTNAME)
```

```
C

      IF (LFPLTNAM.EQ.0) FPLTNAME = 'LPAPER'
      LFPLTNAM = LENGCHAR(FPLTNAME)

c     write(15,*)'genfplot 02',fpltname(1:lfpltnam)

      OPEN(75,FILE=FPLTNAME(1:LFPLTNAM))
      CLOS (75,STATUS='DELETE')
      OPEN(75,FILE=FPLTNAME(1:LFPLTNAM))

C
C     READ HEAD OF FILE

      READ(77,27)HEADRPIX,HEADRPIY,HEADXMAX,HEADYMAX,
     X           HEADXHI ,HEADYHI ,HEADLENG,HEADRESV
27    FORMAT(6I10,2I10)
c     write(15,*)'GENFPLT LENG',HEADLENG

      READ(77,7)ADUMMY
7     FORMAT(A1)
      IXP = INTS(0)
      IYP = INTS(0)

      DO 100 J = 1,HEADLENG

c       write(15,*)'GENFPLT PLOT>',IX,J,IY,IPEN

        READ(77,37)IX,IY
37      FORMAT(2I5)
        IPEN = 2
        IF (IY.LT.INTS(0)) IPEN = 3
        IY     = INTS(IABS(IY))
        IY     = IY - INTS(1) /* IN PLOT ADD 1 FOR -0 = 0 SO RECORRECT Y
        IF (IPEN.EQ.3) GOTO 200
        CALL BDRWLINE(IXP,IYP*INTS(2),IX,IY*INTS(2))
c        CALL FDRWLINE(IXP,IYP*INTS(2),IX,IY*INTS(2))
200     IXP = IX
        IYP = IY
100   CONTINUE
c     write(15,*)'genfplot pass loop'

C
C     GENERATE PLOTTED FILE

      ITEMP1 = 5
      ITEMP1 = LS(ITEMP1,INTS(8))
      ITEMP2 = 5
      IGCONTRO = OR(ITEMP1,ITEMP2)

      IHEADFOR = 12
      WRITE(75,275)HEADFORM(2:2)
275   FORMAT(A1)

      DO 420 I = 1,120
      LPAPER(I:I) = '*'
```

```
C


420     CONTINUE
        WRITE(75,*)LPAPER
        WRITE(75,*)LPAPER

        DO 400 I = 1,1
        IHEADFOR = 10
        WRITE(75,275)HEADFORM(2:2)
400     CONTINUE
        WRITE(75,375)'GRAPHIC  OUTPUT'
375     FORMAT(50X,A15)
        DO 410 I = 1,1
        IHEADFOR = 10
        WRITE(75,275)HEADFORM(2:2)
410     CONTINUE

        WRITE(75,*)LPAPER
        WRITE(75,*)LPAPER

C
C       WRITE INTO FILE

        DO 1000 J = 1,600
c        write(15,*)'genfplot write into file',J

          DO 2000 K =1 , 120
            ITEMP1 = INTS(ICHAR( PAPER(J)(K:K) ) )
            ITEMP5 = :000000
            MARK   = :000040
            TESTMK = :000001

            DO 3000 L=1,6
              ITEMP3 = AND(ITEMP1,TESTMK)
              IF (ITEMP3.NE.INTS(0)) ITEMP5 = OR(ITEMP5,MARK)
              TESTMK = LS(TESTMK,INTS(1))
              MARK   = RS(MARK,INTS(1))
3000        CONTINUE

            ITEMP5 = OR(ITEMP5,INTS(:000100))
            LPAPER2(K:K) = CHAR(INTL(ITEMP5))
2000      CONTINUE

        WRITE(75,175)GCONTROL(1:1),LPAPER2
175     FORMAT(A1,A120)
1000    CONTINUE

        WRITE(75,*)LPAPER
        WRITE(75,*)LPAPER

        DO 800 I = 1,1
        IHEADFOR = 10
c        WRITE(75,275)HEADFORM(2:2)
800     CONTINUE

c        WRITE(75,*)LPAPER
c        WRITE(75,*)LPAPER

        CLOSE(77)
        CLOSE(75)
```

```
C

9999  CONTINUE
      RETURN
      END


C
C     GENERATE VECTOR ON TERMINAL
C
      SUBROUTINE GENTPLOT

      INTEGER*2   LENGCHAR, LENGLINE, LFPLTNAM, LFVECNAM
      INTEGER*2   IX, IY, IXP, IYP
      INTEGER*2   HEADRPIX, HEADRPIY, HEADXMAX, HEADYMAX
      INTEGER*2   HEADXHI  , HEADYHI
      INTEGER*4   HEADLENG, HEADRESV
      INTEGER*4   IPEN
      CHARACTER*30 FPLTNAME, LINEBUFF, FVECNAME

      INTEGER*2       ITEMP1, ITEMP2, ITEMP3, ITEMP4, ITEMP5, ITEMP6
      INTEGER*2       MARK, TESTMK, INFRCH
      INTEGER*2       IHEADFOR, IGCONTRO
      CHARACTER*4     FILEBUFF
      CHARACTER*1     ADUMMY
      CHARACTER*2     HEADFORM, GCONTROL, CHFRIN
      CHARACTER*120 LPAPER, LPAPER2
      EQUIVALENCE     (IHEADFOR, HEADFORM), (IGCONTRO, GCONTROL)
      EQUIVALENCE     (INFRCH, CHFRIN)

$INSERT IMGPAPER.F77

C
C     CLEAR PAPER
      DO 1100 J1 = 1,600
        DO 1200 J2 = 1,120
          PAPER(J1)(J2:J2) = CHAR(:140300)
1200    CONTINUE
1100  CONTINUE


C
C     OPEN VECTOR FILE

      LINEBUFF = 'ENTER VECTOR FILENAME
      LENGLINE = LENGCHAR(LINEBUFF)+1
      CALL TNOUA(LINEBUFF, LENGLINE)
      READ(1,17)FPLTNAME
17    FORMAT(A30)
      LFPLTNAM = LENGCHAR(FPLTNAME)
      IF (LFPLTNAM.EQ.0) FPLTNAME = 'VECTOR'
      LFPLTNAM = LENGCHAR(FPLTNAME)

      FVECNAME = FPLTNAME
      LFVECNAM = LFPLTNAM

c      write(15,*)'genfplot 01', FVECname(1:LFVECnam)
      OPEN(77,FILE=FVECNAME(1:LFVECNAM))

      call entgrp
```

```
C
C
C
C       READ HEAD OF FILE
C
        READ(77,27)HEADRPIX,HEADRPIY,HEADXMAX,HEADYMAX,
       X            HEADXHI  ,HEADYHI  ,HEADLENG,HEADRESV
27      FORMAT(6I10,2I10)
c        write(15,*)'GENTPLOT LENG',HEADLENG
C
        READ(77,7)ADUMMY
7       FORMAT(A1)
C
        IXP = INTS(0)
        IYP = INTS(0)
        DO 100 J = 1,HEADLENG
          READ(77,37)IX,IY
37        FORMAT(2I5)
          IPEN = 2
          IF (IY.LT.INTS(0)) IPEN = 3
          IY     = INTS(IABS(IY))
          IY     = IY - INTS(1) /* IN PLOT ADD 1 FOR -0 = 0 SO RECORRECT Y
c          write(15,*)'GENFPLT PLOT>',IX,J,IY,IPEN
          IF (IPEN.EQ.3) GOTO 200
          CALL TBDRWLINE(IXP,IYP,IX,IY)
200       IXP = IX
          IYP = IY
100     CONTINUE
        CLOSE(77)


        goto 9999


9999    CONTINUE
        call tiin(inkey)
        call clrtek
        call enttxt
        RETURN
        END


C
C       DUMP FROM TERMINAL TO LOGICAL PAPER (FILE)
C
        SUBROUTINE HARDCOPY

        INTEGER*2     LENGCHAR,IROW1,ITEMP1,ITEMP2,TERMINP(80),IHEADFOR
        INTEGER*2     IGCONTRO,ITEMP3,ITEMP4,ITEMP5,ITEMP6,MARK,TESTNK
        INTEGER*2     LFPLTNAM,LENGLINE
        CHARACTER*2   GCONTROL,HEADFORM
        CHARACTER*3   BROW1,ROW1
        CHARACTER*30  DUMPTERM,FPLTNAME,LINEBUFF
        CHARACTER*120 LPAPER
        EQUIVALENCE   (LPAPER,TERMINP),(IGCONTRO,GCONTROL)
        EQUIVALENCE   (IHEADFOR,HEADFORM)

C
C       HIND GRAPHIC SCREEN & ENTER TEXT MODE

        CALL ENTNAT
        GALL SELSCR(INTS(0))
```

```
C
      CALL ENTTEK
      CALL ENTTXT
C
C     OPEN LOGICAL PAPER FILE

      LINEBUFF = 'ENTER PLOTTED FILENAME :
      LENGLINE = LENGCHAR(LINEBUFF)+INTS(1)
      CALL TNOUA(LINEBUFF,LENGLINE)
      READ(1,17)FPLTNAME
17    FORMAT(A30)
      LFPLTNAM = LENGCHAR(FPLTNAME)
      IF (LFPLTNAM.EQ.INTS(0)) FPLTNAME='LPAPER'
      LFPLTNAM = LENGCHAR(FPLTNAME)
      OPEN(75,FILE=FPLTNAME(1:LFPLTNAM))
      CLOSE(75,STATUS='DELETE')
      OPEN(75,FILE=FPLTNAME(1:LFPLTNAM))
      CALL ENTGRP
      CALL ENTNAT
      CALL SELSCR(INTS(3))

      ITEMP1 = 5
      ITEMP1 = LS(ITEMP1,INTS(8))
      ITEMP2 = 5
      IGCONTRO = OR(ITEMP1,ITEMP2)

C
C     LOOP TO READ EACH ROW OF TERMINAL

      IHEADFOR = 12
      WRITE(75,275)HEADFORM(2:2)
275   FORMAT(A1)

      DO 420 I = 1,120
      LPAPER(I:I) = '*'


420   CONTINUE
      WRITE(75,*)LPAPER
      WRITE(75,*)LPAPER

      DO 400 I = 1,3
      IHEADFOR = 10
      WRITE(75,275)HEADFORM(2:2)
400   CONTINUE
      WRITE(75,375)'GRAPHIC  OUTPUT'
375   FORMAT(60X,A15)
      DO 410 I = 1,4
      IHEADFOR = 10
      WRITE(75,275)HEADFORM(2:2)
410   CONTINUE
      WRITE(75,*)LPAPER
      WRITE(75,*)LPAPER

      DO 500 IROW1 = 0,299
      DUMPTERM = '
      WRITE(BROW1,100)IROW1
100   FORMAT(I3)
      CALL DELBLANK(BROW1,ROW1)
      ITEMP1 = LENGCHAR(ROW1)
      DUMPTERM = CHAR(:233)//'D'//ROW1(1:ITEMP1)//';1;1#'        /* dump pa
```

```
C

      ITEMP1 = LENGCHAR(DUMPTERM)
      CALL TNOUA (DUMPTERM, ITEMP1)

C
C     WRITE INTO FILE

      DO 8000 I =1 , 60
        CALL T1IN  (ITEMP1)
        ITEMP5 = :000000
        MARK   = :000040
        TESTMK = :000001
        DO 6000 J=1,6
          ITEMP3 = AND(ITEMP1, TESTMK)
          IF (ITEMP3. NE. INTS(0)) ITEMP5 = OR(ITEMP5, MARK)
          TESTMK = LS(TESTMK, INTS(1))
          MARK   = RS(MARK, INTS(1))
6000    CONTINUE
        ITEMP5 = OR(ITEMP5, INTS(:000100))
        ITEMP5 = LS(ITEMP5, INTS(8))

        CALL T1IN  (ITEMP2)
        ITEMP6 = :000000
        MARK   = :000040
        TESTMK = :000001
        DO 7000 J=1,6
          ITEMP4 = AND(ITEMP2, TESTMK)
          IF (ITEMP4. NE. INTS(0)) ITEMP6 = OR(ITEMP6, MARK)
          TESTMK = LS(TESTMK, INTS(1))
          MARK   = RS(MARK, INTS(1))
7000    CONTINUE
        ITEMP6 = OR(ITEMP6, INTS(:000100))
        ITEMP6 = AND(ITEMP6, INTS(:000377))
        ITEMP5 = OR(ITEMP5, ITEMP6)
        TERMINP(I)=ITEMP5
8000  CONTINUE

      WRITE(75, 175)GCONTROL(1:1), LPAPER
175   FORMAT(A1, A120)

500   CONTINUE

      IHEADFOR = 10
      WRITE(75, 275)HEADFORM(2:2)
      DO 450 I = 1, 120
      LPAPER(I:I) = '*'


450   CONTINUE
      WRITE(75, *)LPAPER
      WRITE(75, *)LPAPER

C
C     END OF DUMP

      CLOSE (75)
      CALL ENTTEK
      RETURN
      END
```

```
C
C
C
C
      SUBROUTINE DUMPFILE

      INTEGER*2  LENGCHAR, LENGLINE, LFPLTNAM, IX, IY, ITEMP1
      INTEGER*2  HEADRPIX, HEADRPIY, HEADXMAX, HEADYMAX
      INTEGER*2  HEADXHI  , HEADYHI  , HEADLENG, HEADRESV
      INTEGER*4  IPEN
      CHARACTER*30 FPLTNAME, LINEBUFF
      CHARACTER*1  ADUMMY

      LINEBUFF = 'ENTER VECTOR FILENAME :
      LENGLINE = LENGCHAR(LINEBUFF)+1
c      write(15,*)'dumpfile', lengline
      CALL TNOUA(LINEBUFF, LENGLINE)
      READ(1,17)FPLTNAME
17    FORMAT(A30)
      LFPLTNAM = LENGCHAR(FPLTNAME)
      IF (LFPLTNAM.EQ.0) FPLTNAME = 'VECTOR'
      LFPLTNAM =LENGCHAR(FPLTNAME)

c      write(15,*)'dumpfile', fpltname(1:lfpltnam), '*'
      OPEN(77, FILE=FPLTNAME(1:LFPLTNAM))
c      write(15,*)'dumpfile', '***'
      READ(77,27)HEADRPIX, HEADRPIY, HEADXMAX, HEADYMAX,
     X           HEADXHI  , HEADYHI  , HEADLENG, HEADRESV
27    FORMAT(6A2, 2A4)
c      write(15,*)'DUMPFILE LENG', HEADLENG

      READ(77,7)ADUMMY
7     FORMAT(A1)

      DO 100 J = 1, INTL(HEADLENG)
c        write(15,*)'DUMPFILE', J
        READ(77,37)IX, IY
37      FORMAT(2A2)
c        write(15,*)'DUMPFILE', IX, IY
        IPEN = 2
        IF (ITEMP1.LT.0) IPEN = 3
        IY     =  INTS(IABS(IY)-1)
c        write(15,*)'DUMPFILE PLOT', IX, IY, IPEN
100   CONTINUE

      RETURN
      END
```

```
User: ECPB817203                                -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>GRAPHIC.F77

**********************************************************************
**********************************************************************

WWWWW  WWW  WWWW  WWWW   WWW    W   WWWWW  WWW   WWW   WWW
W     W   W W   W W   W W   W W   W WW      W W   W W   W W   W
W     W     W   W W   W W   W W   W W        W     W W   W     W
WWWW  W     WWWW  WWWW   WWW    W   W      W   W W W   WW
W     W     W     W   W W   W    W   W    W     W   W   W   W
W     W   W W     W   W W   W    W   W    W     W   W W W   W
WWWWW  WWW  W     WWWW   WWW  WWW  W     WWWWW  WWW   WWW

 WWW  WWWW   WWW  WWWW  W   W WWW   WWW
W   W W   W W   W W   W W   W W   W W  W
W     W   W W   W W   W W   W W   W W
W     WWWW  WWWW  WWWW  WWWWW W   W WWW
W  WW W  W W   W W   W W   W W   W W  W
W   W W  W  W   W W   W W   W W   W W  W
 WWW  W   W W   W W   W W   W WWW  WWW

**********************************************************************
**********************************************************************

Label: PRT012 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>GRAPHIC.F77
File last modified: 87-08-27.14:34:20.Thu

Spooled:  87-09-18.10:37:32.Fri   [Spooler rev 19.4]
Started:  87-09-18.10:51:44.Fri   on: PRO        by: PRO
```

```
C
C
C
C     DRAW LINE BY TREKTRONIX MODE
C
      SUBROUTINE  TDRWLINE(IX1,IY1,IX2,IY2)

      INTEGER*2    IX1,IY1,IX2,IY2,IBASE
      INTEGER*4    ITX1,ITY1,ITX2,ITY2
      INTEGER*2    ITEMP1,ITEMP2,ITEMP3,ITEMP4
      REAL*8       XTEMP1,XTEMP2,XTEMP

      XTEMP  = IX1
      XTEMP1 = (XTEMP*1024)/720                    /*  1024/720 = 1.42

      XTEMP  = IX2
      XTEMP2 = (XTEMP*1024)/720

      ITX1   = IDINT(XTEMP1+.00001)
      ITX2   = IDINT(XTEMP2+.00001)
      IBASE = 2

      ITEMP1 = (IY1+1)/3
      ITEMP2 = (ITEMP1*8)-IBASE
      ITEMP3 = ((IY1+1)-(ITEMP1*3))*3
      ITY1   = ITEMP2+ITEMP3
      IF (ITY1.EQ.774) ITY1 = 775

      ITEMP1 = (IY2+1)/3
      ITEMP2 = (ITEMP1*8)-IBASE
      ITEMP3 = ((IY2+1)-(ITEMP1*3))*3
      ITY2   = ITEMP2+ITEMP3
      IF (ITY2.EQ.774) ITY2 = 775

      CALL GRAPHIC (3,ITX1,ITY1)
      CALL GRAPHIC (4,ITX2,ITY2)

      RETURN
      END

C
C     SUBROUTINE XDRWLINE
C
      SUBROUTINE XDRWLINE(IX1S,IY1S,IX2S,IY2S)

      INTEGER*2 IX1S,IY1S,IX2S,IY2S
      INTEGER*4 IX1,IY1,IX2,IY2

      IX1=INTL(IX1S)
      IY1=INTL(IY1S)
      IX2=INTL(IX2S)
      IY2=INTL(IY2S)

      CALL GRAPHIC(3,IX1,IY1)
      CALL GRAPHIC(4,IX2,IY2)
      RETURN
      END

C
C     GRAPHIC
C        FN = 1  ENTER GRAPHIC MODE
C        FN = 2  ENTER TEXT MODE
```

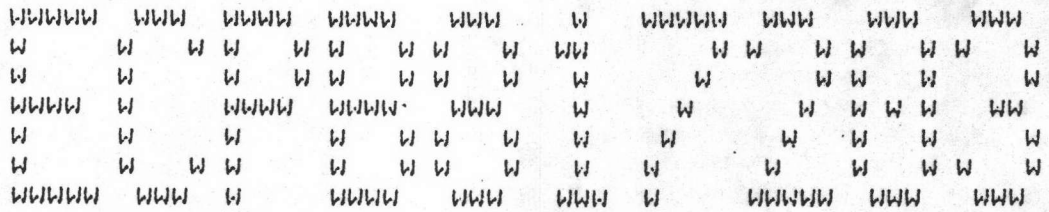
```
C
C       FN = 3  SET SOLID LINE
C       FN = 4  PLOT
C       FN = 5  CLEAR GRAPHIC
C       FN = 6  CLEAR TEXT
C       FN = 7  SIZEL
C       FN = 8  SIZES
C       FN = 9  SET LONG DASHED
C       FN = 10 SET DOT DASHED
C       FN = 11 SET DOT LINE
C       FN = 12 DRAW POINT
C
      SUBROUTINE GRAPHIC(FN, X1, Y1)
      INTEGER*4 FN, X1, Y1
      CHARACTER   ESC, QUES, CNTRL, DOL, SIX, EIGHT, SEMI, BACKQU, VECTOR, ALEK
      CHARACTER   CNTRUN, BUF(4), FS
      CHARACTER*2 CLEARS, CLRGRP, SOLID, DOTTED, ALPHA, SIZEL, SIZES, DDASH
     +          , LDASH
      CHARACTER*3 ENGRPH, ENTEXT
      CHARACTER*7 CUR
C
      ESC    = CHAR(27)
      QUES   = CHAR(63)
      CNTRL  = CHAR(12)
      DOL    = CHAR(36)
      SIX    = CHAR(54)
      EIGHT  = CHAR(56)
      SEMI   = CHAR(59)
      BACKQU = CHAR(96)
      VECTOR = CHAR(29)
      ALEK   = CHAR(97)
      CNTRUN = CHAR(31)
      FS     = CHAR(28)
      CLEARS = ESC//QUES                    /* CLEAR SCREEN TEXT MODE
      CLRGRP = ESC//CNTRL                   /* CLEAR SCREEN GRAPHIC MODE
      ENGRPH = ESC//DOL//SIX                /* ENTER GRAPHIC MODE
      ENTEXT = ESC//DOL//QUES               /* ENTER TEXT MODE
      SOLID  = ESC//BACKQU                  /* SOLID LINE
      DOTTED = ESC//ALEK                    /* DOTTED LINE
      LDASH  = ESC//'d'                     /* LONG DASHED
      DDASH  = ESC//'b'                     /* DOT DASHED
      ALPHA  = ESC//CNTRUN                  /* ALPHA MODE
      SIZEL  = ESC//EIGHT                   /* LARGEST SIZE
      SIZES  = ESC//SEMI                    /* SMALLEST SIZE
      CUR    = ESC//CHAR(91)//CHAR(50)//CHAR(50)//CHAR(59)//CHAR(49)//
     +         CHAR(102)
       IF (X1.GT.1023) X1 = 1023        /* CHECK POSITION
       IF (X1.LT.0) X1 = 0
       IF (Y1.GT.779) Y1 =779
       IF (Y1.LT.0) Y1 = 0

C
       GO TO (10,20,30,40,50,60,80,90,30,30,30,100) FN
   10  CALL TNOUA( ENGRPH, INTS(3))
       GO TO 70
   20  CALL TNOUA (ENTEXT, INTS(3))
       GO TO 70
   30  CALL TNOUA (ALPHA, INTS(2))
       CALL TNOUA (VECTOR, INTS(1))
       IF (FN.EQ.3) CALL TNOUA (SOLID, INTS(2))
```

```
C
      IF (FN.EQ.9) CALL TNOUA (LDASH,INTS(2))
      IF (FN.EQ.10) CALL TNOUA (DDASH,INTS(2))
      IF (FN.EQ.11) CALL TNOUA (DOTTED,INTS(2))
   40 CALL PLOT20 (X1,Y1,BUF)
      CALL TNOUA (BUF,INTS(4))
      GO TO 70
   50 CALL TNOUA (CLRGRP,INTS(2))
      GO TO 70
   60 CALL TNOUA (CLEARS,INTS(2))
      GO TO 70
   80 CALL TNOUA (ALPHA,INTS(2))
      CALL TNOUA (CHAR(9),INTS(1))
      CALL TNOUA (SIZEL,INTS(2))
      GOTO 70
   90 CALL TNOUA (ALPHA,INTS(2))
      CALL TNOUA (CHAR(9),INTS(1))
      CALL TNOUA (SIZES,INTS(2))
      GOTO 70
  100 CALL TNOUA (FS,INTS(1))
   70 RETURN
      END

      SUBROUTINE PLOT20 (X,Y,BUF)
      INTEGER*4  X,Y
      CHARACTER BUF*(*)
      BUF(3:3) = CHAR ( OR (RS(X,5),:40))
      BUF(4:4) = CHAR ( OR (AND(X,:37),:100))
      BUF(1:1) = CHAR ( OR (RS(Y,5),:40))
      BUF(2:2) = CHAR ( OR (AND(Y,:37),:140))
      RETURN
      END
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>INIT.F77

****
****



****
****

Label: PRT013 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>INIT.F77
File last modified: 87-09-04.11:39:44.Fri

Spooled: 87-09-18.10:37:48.Fri [Spooler rev 19.4]
Started: 87-09-18.10:52:20.Fri on: PRO by: PRO

```
C
C
C
C
C
      SUBROUTINE PLOTS(D1,D2,D3)
      INTEGER*4 D1,D2,D3

      CALL INITPARA

      RETURN
      END
C
C
C
      SUBROUTINE INITPARA

$INSERT LGDEVICE.F77
$INSERT HEADFILE.F77
$INSERT INGPAPER.F77

      INTEGER*2     LENGLINE,CHIN,LENGCHAR,OUTKEY,ERRKEY
      CHARACTER*80 LINEBUFF
      CHARACTER*1   INKEY

5     FORMAT(A1)
      DEVKEY = 2                                          /*DEFAULT
10    LINEBUFF='SELECT PLOT MODE T=TERMINAL ,F=FILE : '
      LENGLINE=LENGCHAR(LINEBUFF)+1
      CALL TNOUA (LINEBUFF,LENGLINE)
      READ(1,5)INKEY
      WRITE(1,*)
      IF ((INKEY.EQ.'F').OR.(INKEY.EQ.'f')) GOTO 20
      IF ((INKEY.NE.'T').AND.(INKEY.NE.'t')) GOTO 10

      OUTDEV = 0
      OUTKEY = 0
      DEVKEY = 0
15    LINEBUFF='SELECT PLOT MODE B=BUFFER ,I=IMMEDIATELY :
      LENGLINE=LENGCHAR(LINEBUFF)+1
      CALL TNOUA (LINEBUFF,LENGLINE)
      READ(1,5)INKEY
C      WRITE(1,*)
      IF ((INKEY.EQ.'I').OR.(INKEY.EQ.'i')) GOTO 25
      IF ((INKEY.NE.'B').AND.(INKEY.NE.'b')) GOTO 15
      INTERACT = .FALSE.
      GOTO 30
25    INTERACT = .TRUE.
      GOTO 30

20    CONTINUE
      INTERACT = .FALSE.
      FILEPLOT = .TRUE.
      OUTKEY = 3
      OUTDEV = 77
      LINEBUFF='ENTER VECTOR FILENAME :
      LENGLINE=LENGCHAR(LINEBUFF)+1
      CALL TNOUA (LINEBUFF,LENGLINE)
      READ(1,17)FPLTNAME
17    FORMAT(A30)
      LFPLTNAM = LENGCHAR(FPLTNAME)
      IF (LFPLTNAM.EQ.0) FPLTNAME = 'VECTOR'
```

```
C
      LFPLTNAM = LENGCHAR(FPLTNAME)
      GOTO 30

30    CONTINUE
40    WRITE(1,*)
      WRITE(1,*)'SCREEN HAS 700 X 1000 DOTS
      WRITE(1,*)'       1.    5 RPI'
      WRITE(1,*)'       2.   10 RPI'
      WRITE(1,*)'       3.   15 RPI'
      WRITE(1,*)'       4.   25 RPI'
      WRITE(1,*)'       5.   50 RPI'
      WRITE(1,*)'       6.  100 RPI'
      WRITE(1,*)'       7.  200 RPI'
      WRITE(1,*)'       8.  400 RPI'
      WRITE(1,*)'       9.  800 RPI'
      LINEBUFF=' SELECT : '
      LENGLINE=LENGCHAR(LINEBUFF)+1
      CALL TNOUA (LINEBUFF,LENGLINE)
      READ(1,5)INKEY
c      WRITE(1,*)
      IF (INKEY.EQ.'1') RPI =   5
      IF (INKEY.EQ.'2') RPI =  10
      IF (INKEY.EQ.'3') RPI =  15
      IF (INKEY.EQ.'4') RPI =  25
      IF (INKEY.EQ.'5') RPI =  50
      IF (INKEY.EQ.'6') RPI = 100
      IF (INKEY.EQ.'7') RPI = 200
      IF (INKEY.EQ.'8') RPI = 400
      IF (INKEY.EQ.'9') RPI = 800
      RPIX = RPI
      RPIY = RPI
      IF ((INKEY.LT.'1').OR.(INKEY.GT.'9')) GOTO 40

60    CONTINUE
      CONTINUE
      ERRDEV = 76
      ERRKEY = 3
      LINEBUFF='ENTER ERROR OUTPUT FILENAME :
      LENGLINE=LENGCHAR(LINEBUFF)+1
      CALL TNOUA (LINEBUFF,LENGLINE)
      READ(1,17)FERRNAME
      LFERRNAM = LENGCHAR(FERRNAME)
      IF (LFERRNAM.EQ.0) FERRNAME = 'ERROR'
      LFERRNAM = LENGCHAR(FERRNAME)
      GOTO 80

80    CONTINUE
      LINEBUFF='SELECT AXIS MODE H=HORIZONTAL ,V=VERTICAL :
      LENGLINE=LENGCHAR(LINEBUFF)+1
      CALL TNOUA (LINEBUFF,LENGLINE)
      READ(1,5)INKEY
c      WRITE(1,*)
      IF ((INKEY.EQ.'H').OR.(INKEY.EQ.'h')) GOTO 90
      IF ((INKEY.NE.'V').AND.(INKEY.NE.'v')) GOTO 80
      VERTPLOT = .TRUE.
      GOTO 100
90    VERTPLOT = .FALSE.
      GOTO 100

100   CONTINUE
```

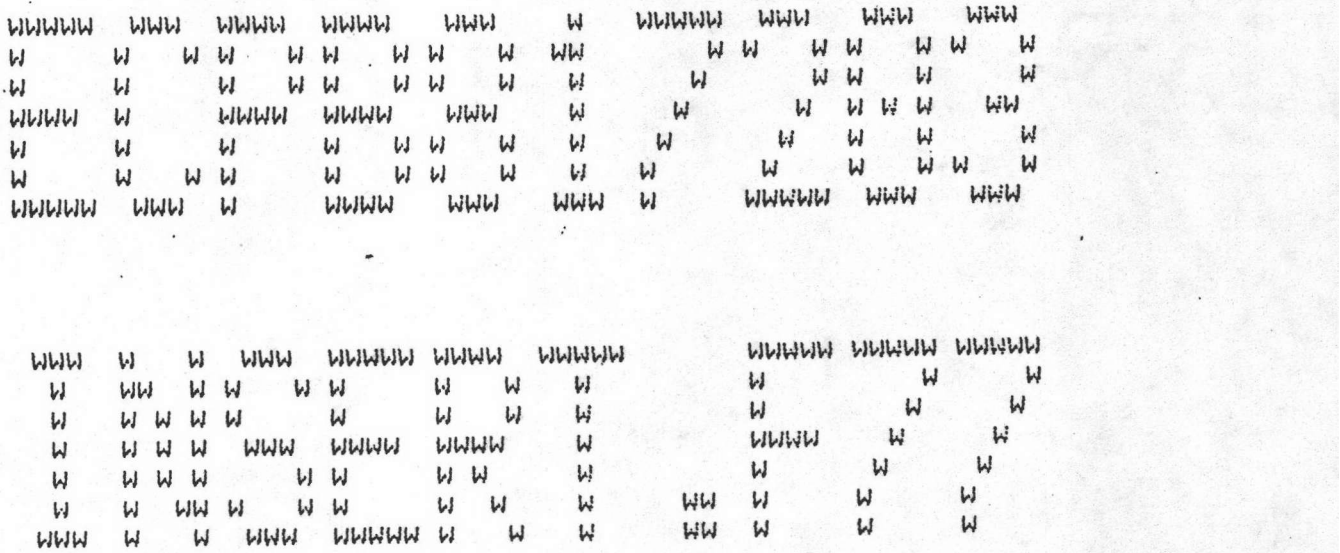
```
C

      CALL INITSYST(OUTKEY,OUTDEV,ERRKEY,ERRDEV,DEVKEY,INTS(0))
      RETURN

      END

C
C
C
      SUBROUTINE  INITSYST(OUTKEY,OUTDEV1,ERRKEY,ERRDEV1,DEVKEY,RESERVE)

$INSERT LGDEVICE.F77
$INSERT HEADFILE.F77
$INSERT IMGPAPER.F77

      INTEGER*2   OUTKEY,OUTDEV1,ERRKEY,ERRDEV1,DEVKEY,RESERVE

      XORG   = 0
      YORG   = 0
      CURX   = 0
      CURY   = 0
      LASTX  = 0
      LASTY  = 0
      SFACTR = 1.
C     RPI    -> RPIX
C        .   -> RPIY .
      OUTDEV = OUTDEV1
      ERRDEV = ERRDEV1
      OUTBUFPT = 1
C     OUTBUF   NEEDN'T INIT
      TCOS     = 1
      TSIN     = 0
      NPLT     = 0
      LNWT     = 1
      SLANT    = 0
      MIRROR   = 0
C     VERTPLOT   RECEIVE DATA FROM KEYBOARD
C     INTERACT
C     FPLTNAME
C     FERRNAME
C     LFPLTNAM   RECEIVE DATA FROM EXECUTION
C     LFERRNAM
      IXCURS   = 0
      IYCURS   = 0
      DO 110 J=1,600
      DO 120 I=1,120
        PAPER(J)(I:I)=CHAR(:140300)
120   CONTINUE
110   CONTINUE

      HEADXMAX = 1023
      HEADYMAX = 759

      IF (OUTKEY.NE.0) GOTO 20
      CALL ENTGRP
      CALL GLRTEK
      CALL PLOT(0.,0.,3)
      GOTO  9999

20    CONTINUE
      OPEN(OUTDEV,FILE=FPLTNAME(1:LFPLTNAM))
```

```
      CLOSE(OUTDEV,STATUS='DELETE')
      OPEN(OUTDEV,FILE=FPLTNAME(1:LFPLTNAM))
      HEADLENG = 0
      HEADXHI  = 0
      HEADYHI  = 0
      HEADRESV = -1
C     WRITE HEAD OF FILE
      WRITE(OUTDEV,17)HEADRPIX,HEADRPIY,HEADXMAX,HEADYMAX,
     X                HEADXHI,HEADYHI,HEADLENG,HEADRESV
17    FORMAT(8I10)
      WRITE(OUTDEV,27)0,-1
27    FORMAT(15HTTTTTTTTTTTTTTT,2I5)
      CALL PLOT(0.,0.,3)
      GOTO 9999

9999  CONTINUE
      OPEN(ERRDEV,FILE=FERRNAME(1:LFERRNAM))
      CLOSE(ERRDEV,STATUS='DELETE')
      OPEN(ERRDEV,FILE=FERRNAME(1:LFERRNAM))
      RETURN
      END

C
C     CLEAR OUTPUT BUFFER TO OUTPUT FILE & TERMINAL
C
      SUBROUTINE CLRBUF

$INSERT LGDEVICE.F77
$INSERT HEADFILE.F77
$INSERT IMGPAPER.F77

      EQUIVALENCE (INFRCH,CHFRIN)

      INTEGER*2 LOOP1,IX,IY,ITEMP1


      INTEGER*2   INFRCH
      CHARACTER*2 CHFRIN
      CHARACTER*4 RECBUF

      IF(OUTBUFPT.LE.1)  GO TO 30                  /* INVALID
      IF(.NOT.FILEPLOT)  GO TO 20                  /* --> 20 FOR TERMINAL OUTPUT

C
C     FILE OUTPUT

c      write(15,*)'clrbuf',outbufpt-1
      DO 40 LOOP1 = 1,OUTBUFPT-1
        WRITE(OUTDEV,17)OUTBUF(1,LOOP1),OUTBUF(2,LOOP1)
        ITEMP1 = INTS(2)
        IF (OUTBUF(2,LOOP1).LT.INTS(0)) ITEMP1=INTS(3)
C        write(15,*)'clrbuf',outbuf(1,LOOP1),LOOP1
C     X             ,outbuf(2,LOOP1),ITEMP1

17      FORMAT(2I5)
40    CONTINUE
      HEADLENG = HEADLENG + OUTBUFPT - 1           /* UPDATE FILE LENGTH
c      write(15,*)'clrbuf headleng',HEADLENG
      GOTO 30
```

```
C
C
C       TERMINAL OUTPUT

20      CONTINUE
        DO 50 LOOP1 = 1,OUTBUFPT-1                          /* X COORD (IN RASTERS)
          IX=OUTBUF(1,LOOP1)                                /* Y COORD
          IY=INTS(IABS(OUTBUF(2,LOOP1)))
          IY=IY - INTS(1)  /* IN PLOT ADD 1 FOR -0 = 0 SO  RECORRECT
          IF(OUTBUF(2,LOOP1).LT.0) GOTO 60
c          write(15,*)'before drwline clrbuf',IXCURS,IYCURS,IX,IY
          CALL XDRWLINE(IXCURS,IYCURS,IX,IY)
60        IXCURS=IX
          IYCURS=IY
50      CONTINUE
        GO TO 30

30      CONTINUE                                    /* RESET BUFFER POINTER
        OUTBUFPT=1
        RETURN
        END

C
C
C
        SUBROUTINE  ENDPLT

$INSERT LGDEVICE.F77
$INSERT HEADFILE.F77
$INSERT IMGPAPER.F77

        REAL*4 XEND,YEND
c        write(15,*)'endplt start'
        CALL CLRBUF
        XEND = 0.
        YEND = 0.
        CALL PLOT (XEND,YEND,3)          /* LEFT CURSOR AT ORIGIN

        IF (FILEPLOT) GOTO 20
        CALL CLRTEK                              /* RETURN TERMINAL TO NORMAL MODE
        CALL ENTTXT
        GOTO 9999

20      CONTINUE
        DO 25  J = 1,10
          WRITE(OUTDEV,27)0,-1
25      CONTINUE
27      FORMAT(2I5)
        REWIND(OUTDEV)
        WRITE(OUTDEV,17)HEADRPIX,HEADRPIY,HEADXMAX,HEADYMAX,
                        HEADXHI  ,HEADYHI  ,HEADLENG,HEADRESV
       X
17      FORMAT(8I10)
        CLOSE(OUTDEV)
        GOTO 9999

9999    CONTINUE
        CLOSE(ERRDEV)
c        write(15,*)'endplt pass'
        RETURN
        END
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>INSERT.F77



Label: PRT016 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>INSERT.F77
File last modified: 87-09-08.14:15:40.Tue

Spooled: 87-09-18.10:37:52.Fri [Spooler rev 19.4]
Started: 87-09-18.10:57:00.Fri on: PRO by: PRO

```
C
C
C
C       IMAGE PAPER FOR TRACE
C
        COMMON /PAPER/
       X         PAPER,
       X         IXCURS,
       X         IYCURS

        CHARACTER*120 PAPER(600)
        INTEGER*2     IXCURS, IYCURS

C
C     DEFINE HEAD PLOT FILE  FORMAT FOR BASIC GRAPHIC SUBROUTINES PACKAGE
C
          INTEGER*2 HEADSIZE
          PARAMETER HEADSIZE=20    /* LENGTH OF  HEAD FILE

          COMMON /HEADFILE/
       X     HEADRPIX,          /* RASTERS PER INCH, X DIRECTION
       X     HEADRPIY,          /* RASTERS PER INCH, Y DIRECTION
       X     HEADXMAX,          /* MAX X COORDINATE ON PLOT DEVICE (RASTERS)
       X     HEADYMAX,          /* MAX Y COORDINATE ON PLOT DEVICE (RASTERS)
       X     HEADXHI,           /* HIGHEST X COORDINATE IN FILE (RASTERS)
       X     HEADYHI,           /* HIGHEST Y COORDINATE IN FILE (RASTERS)
       X     HEADLENG,          /* LENGTH OF HEADFILE
       X     HEADRESV           /* RESERVE FOR FURTHER USE

          INTEGER*2 HEADRPIX, HEADRPIY, HEADXMAX, HEADYMAX, HEADXHI, HEADYHI
          INTEGER*4 HEADLENG, HEADRESV

C
C       GROBAL VARIABLES FILE
C
          NOLIST
          INTEGER*2 OUTBUFSZ
          PARAMETER OUTBUFSZ = 1024    /* MAX NUMBER POINTS IN OUTPUT BUFFER
          COMMON /LGDEVICE/
       X     XORG,              /* X-ORIGIN (RASTERS)
       X     YORG,              /* Y-ORIGIN (RASTERS)
       X     CURX,              /* CURRENT X POSITION (RASTERS)
       X     CURY,              /* CURRENT Y POSITION (RASTERS)
       X     LASTX,             /* LAST VALID X POS (INCHES REL TO ORG)
       X     LASTY,             /* LAST VALID Y POS (INCHES REL TO ORG)
       X     SFACTR,            /* SCALE FACTOR
       X     RPIX,              /* RASTERS PER INCH IN X DIRECTION
       X     RPIY,              /* RASTERS PER INCH IN Y DIRECTION
       X     OUTDEV,            /* OUTPUT DEVICE NUMBER
       X     ERRDEV,            /* ERROR DEVICE NUMBER
       X     FILEPLOT,          /* .TRUE. IF PLOTTING TO FILE
       X     OUTBUFPT,              /* CURRENT INDEX INTO OBUF
       X     OUTBUF(2,OUTBUFSZ),    /* OUTPUT BUFFER
       X     TCOS,                  /* COSINE OF ROTATION VALUE
       X     TSIN,                  /* SINE OF ROTATION VALUE
       X     NPLT,                  /* NO-PLOT FLAG FOR 'NUMBER'
       X     LNWT,                  /* CURRENT LINE WEIGHT
       X     SLANT,                 /* CURRENT CHARACTER SLANT
       X     VERTPLOT,              /* .TRUE. IF PLOTTING VERTICALLY
       X     INTERACT,              /* .TRUE. IF INTERACTIVE PLOTTING
       X     FPLTNAME,              /* PLOT FILENAME
       X     LFPLTNAM,              /* LENGTH OF PLOT FILENAME
```

```
C
X       FERRNAME,                  /* ERROR FILENAME
X       LFERRNAM,                  /* LENGTH OF ERROR FILENAME
X       NARRAY(20),                /* ARRAY FOR GENERAL USE
X       NCHRS,                     /* NUMBER OF USED ARRAY
X       MIRROR                     /* MIRROR FLAG 0-3
      INTEGER*2 XORG, YORG, CURX, CURY, ODEV, EDEV, AMLCLN,
X               OUTBUFPT, OUTBUF, NARRAY, NPLT, NCHRS, LNWT,
X               LFPLTNAM, LFERRNAM, OUTDEV, ERRDEV, MIRROR
      REAL*4    TCOS, TSIN, SFACTR, RPIX, RPIY, LASTX, LASTY, SLANT, COSD, SIND
      LOGICAL   FILEPLOT, VERTPLOT, INTERACT, SKS
      CHARACTER*30  FPLTNAME, FERRNAME
      LIST

C
C     CHARACTER DEFINITIONS FOR BASIC SUBROUTINE PACKAGE
C
      INTEGER*2  SYM(3,2000)
      COMMON /SYM/SYM
      DATA SYM /
     #    /*  *** A *** */
     + 0,   193,    8,2,     0,   80,2,    20,   20,2,   60,    0,
     + 2,    20,  -20,2,     0,  -80,3,  -100,   50,2,  100,    0,
     #    /*  *** B *** */
     + 0,   194,   12,2,     0,  100,2,    80,    0,2,   20,  -20,
     + 2,     0,   -5,2,   -20,  -20,2,   -80,    0,3,   80,    0,
     + 2,    20,  -20,2,     0,  -15,2,   -20,  -20,2,  -80,    0,
     #    /*  *** C *** */
     + 0,   195,    9,3,   100,   80,2,   -20,   20,2,  -60,    0,
     + 2,   -20,  -20,2,     0,  -60,2,    20,  -20,2,   60,    0,
     + 2,    20,   20,
     #    /*  *** D *** */
     + 0,   196,    7,2,     0,  100,2,    80,    0,2,   20,  -20,
     + 2,     0,  -60,2,   -20,  -20,2,   -80,    0,
     #    /*  *** E *** */
     + 0,   197,    7,2,     0,  100,2,   100,    0,3,  -100,  -50,
     + 2,    75,    0,3,   -75,  -50,2,   100,    0,
     #    /*  *** F *** */
     + 0,   198,    5,2,     0,  100,2,   100,    0,3,  -100,  -50,
     + 2,    75,    0,
     #    /*  *** G *** */
     + 0,   199,   11,3,   100,   80,2,   -20,   20,2,  -60,    0,
     + 2,   -20,  -20,2,     0,  -60,2,    20,  -20,2,   60,    0,
     + 2,    20,   20,2,     0,   20,2,   -40,    0,
     #    /*  *** H *** */
     + 0,   200,    6,2,     0,  100,3,   100,    0,2,    0, -100,
     + 3,  -100,   50,2,   100,    0,
     #    /*  *** I *** */
     + 0,   201,    7,3,    20,    0,2,   60,    0,3,  -30,    0,
     + 2,     0,  100,3,   -30,    0,2,   60,    0,
     #    /*  *** J *** */
     + 0,   202,    6,3,     0,   20,2,   20,  -20,2,   60,    0,
     + 2,    20,   20,2,     0,   80,
     #    /*  *** K *** */
     + 0,   203,    6,2,     0,  100,3,    0,  -50,2,  100,   50,
     + 3,   -50,  -25,2,    50,  -75,
     #    /*  *** L *** */
     + 0,   204,    4,3,     0,  100,2,    0, -100,2,  100,    0,
     #    /*  *** M *** */
     + 0,   205,    5,2,     0,  100,2,   50,  -40,2,   50,   40,
     + 2,     0, -100,
```

```
#    /*  *** N *** */
+ 0,   205,    4, 2,     0,  100, 2,   100, -100, 2,     0,  100,
#    /*  *** O *** */
+ 0,   207,   10, 3,     0,   20, 2,     0,   60, 2,    20,   20,
+ 2,    60,    0, 2,    20,  -20, 2,     0,  -60, 2,   -20,  -20,
+ 2,   -60,    0, 2,   -20,   20,
#    /*  *** P *** */
+ 0,   208,    7, 2,     0,  100, 2,    80,    0, 2,    20,  -20,
+ 2,     0,  -10, 2,   -20,  -20, 2,   -80,    0,
#    /*  *** Q *** */
+ 0,   209,   12, 3,     0,   20, 2,     0,   60, 2,    20,   20,
+ 2,    60,    0, 2,    20,  -20, 2,     0,  -60, 2,   -20,  -20,
+ 2,   -60,    0, 2,   -20,   20, 3,    80,    0, 2,    20,  -20,
#    /*  *** R *** */
+ 0,   210,    9, 2,     0,  100, 2,    80,    0, 2,    20,  -20,
+ 2,     0,  -10, 2,   -20,  -20, 2,   -80,    0, 3,    60,    0,
+ 2,    40,  -50,
#    /*  *** S *** */
+ 0,   211,   13, 3,   100,   80, 2,   -20,   20, 2,   -60,    0,
+ 2,   -20,  -20, 2,     0,  -10, 2,    20,  -20, 2,    60,    0,
+ 2,    20,  -20, 2,     0,  -10, 2,   -20,  -20, 2,   -60,    0,
+ 2,   -20,   20,
#    /*  *** T *** */
+ 0,   212,    5, 3,     0,  100, 2,   100,    0, 3,   -50,    0,
+ 2,     0, -100,
#    /*  *** U *** */
+ 0,   213,    7, 3,     0,  100, 2,     0,  -80, 2,    20,  -20,
+ 2,    60,    0, 2,    20,   20, 2,     0,   80,
#    /*  *** V *** */
+ 0,   214,    4, 3,     0,  100, 2,    50, -100, 2,    50,  100,
#    /*  *** W *** */
+ 0,   215,    6, 3,     0,  100, 2,    25, -100, 2,    25,   50,
+ 2,    25,  -50, 2,    25,  100,
#    /*  *** X *** */
+ 0,   216,    4, 2,   100,  100, 3,  -100,    0, 2,   100, -100,
#    /*  *** Y *** */
+ 0,   217,    6, 3,     0,  100, 2,    50,  -50, 2,    50,   50,
+ 3,   -50,  -50, 2,     0,  -50,
#    /*  *** Z *** */
+ 0,   218,    5, 3,     0,  100, 2,   100,    0, 2,  -100, -100,
+ 2,   100,    0,
#    /*  *** 0 *** */
+ 0,   176,   12, 3,     0,   20, 2,     0,   60, 2,    20,   20,
+ 2,    60,    0, 2,    20,  -20, 2,     0,  -60, 2,   -20,  -20,
+ 2,   -60,    0, 2,   -20,   20, 3,     0,  -20, 2,   100,  100,
#    /*  *** 1 *** */
+ 0,   177,    6, 3,    40,   80, 2,    10,   20, 2,     0, -100,
+ 3,   -30,    0, 2,    60,    0,
#    /*  *** 2 *** */
+ 0,   178,   10, 3,     0,   80, 2,    20,   20, 2,    60,    0,
+ 2,    20,  -20, 2,     0,  -10, 2,   -20,  -20, 2,   -80,  -30,
+ 2,     0,  -20, 2,   100,    0,
#    /*  *** 3 *** */
+ 0,   179,   14, 3,     0,   80, 2,    20,   20, 2,    60,    0,
+ 2,    20,  -20, 2,     0,  -10, 2,   -20,  -20, 2,   -40,    0,
+ 3,    40,    0, 2,    20,  -20, 2,     0,  -10, 2,   -20,  -20,
+ 2,   -60,    0, 2,   -20,   20,
#    /*  *** 4 *** */
+ 0,   180,    6, 3,    80,    0, 2,     0,  100, 3,    20,  -60,
+ 2,   -70,    0, 2,    20,   60,
```

```
C
#    /*  *** 5 *** */
+ 0,   181,   10,3,     0,   20,2,    20,  -20,2,    60,    0,
+ 2,    20,   20,2,     0,   20,2,   -20,   20,2,   -80,    0,
+ 2,     0,   40,2,   100,    0,
#    /*  *** 6 *** */
+ 0,   182,   15,3,     0,   20,2,    20,  -20,2,    60,    0,
+ 2,    20,   20,2,     0,   10,2,   -20,   20,2,   -60,    0,
+ 2,   -20,  -20,2,     0,  -10,3,     0,   10,2,     0,   50,
+ 2,    20,   20,2,    60,    0,2,    20,  -20,
#    /*  *** 7 *** */
+ 0,   183,    5,3,    10,    0,2,     0,   10,2,    90,   70,
+ 2,  -100,    0,
#    /*  *** 8 *** */
+ 0,   184,   18,3,     0,   20,2,    20,  -20,2,    60,    0,
+ 2,    20,   20,2,     0,   10,2,   -20,   20,2,   -60,    0,
+ 2,   -20,  -20,2,     0,  -10,3,    20,   30,2,   -20,   20,
+ 2,     0,   10,2,    20,   20,2,    60,    0,2,    20,  -20,
+ 2,     0,  -10,2,   -20,  -20,
#    /*  *** 9 *** */
+ 0,   185,   13,3,     0,   20,2,    20,  -20,2,    60,    0,
+ 2,    20,   20,2,     0,   60,2,   -20,   20,2,   -60,    0,
+ 2,   -20,  -20,2,     0,  -10,2,    20,  -20,2,    60,    0,
+ 2,    20,  -20,
#    /*  ***! *** */
+ 0,   161,   10,3,    50,   30,2,   -10,   70,2,    20,    0,
+ 2,   -10,  -70,3,   -10,  -30,2,    20,    0,2,     0,   10,
+ 2,   -20,    0,2,     0,  -10,
#    /*  *** " *** */
+ 0,   162,   11,3,    30,  100,2,    20,    0,2,    -5,  -20,
+ 2,   -10,    0,2,    -5,   20,3,    30,    0,2,    20,    0,
+ 2,    -5,  -20,2,   -10,    0,2,    -5,   20,
#    /*  *** # *** */
+ 0,   163,    9,3,    20,    0,2,    10,  100,3,    50,    0,
+ 2,   -10, -100,3,    30,   30,2,  -100,    0,3,     0,   40,
+ 2,   100,    0,
#    /*  *** $ *** */
+ 0,   164,   17,3,     0,   20,2,    20,  -20,2,    60,    0,
+ 2,    20,   20,2,     0,   10,2,   -20,   20,2,   -60,    0,
+ 2,   -20,   20,2,     0,   10,2,    20,   20,2,    60,    0,
+ 2,    20,  -20,3,   -40,   20,2,     0, -100,3,   -20,    0,
+ 2,     0,  100,
#    /*  *** % *** */
+ 0,   165,   12,2,   100,  100,3,   -80,  -20,2,    20,    0,
+ 2,     0,   10,2,   -20,    0,2,     0,  -10,3,    60,  -60,
+ 2,   -20,    0,2,     0,  -10,2,    20,    0,2,     0,   10,
#    /*  *** & *** */
+ 0,   166,   13,3,   100,    0,2,  -100,   70,2,     0,   10,
+ 2,    20,   20,2,    20,    0,2,    20,  -20,2,   -60,  -30,
+ 2,     0,  -10,2,    20,  -20,2,    20,    0,2,    40,   20,
+ 2,    20,   10,
#    /*  *** ' *** */
+ 0,   167,    5,3,    40,  100,2,    20,    0,2,   -10,  -20,
+ 2,   -10,   20,
#    /*  *** ( *** */
+ 0,   168,    7,3,   100,    0,2,   -20,    0,2,   -20,   20,
+ 2,     0,   60,2,    20,   20,2,    20,    0,
#    /*  *** ) *** */
+ 0,   169,    6,2,    20,    0,2,    20,   20,2,     0,   60,
+ 2,   -20,   20,2,   -20,    0,
#    /*  *** * *** */
```

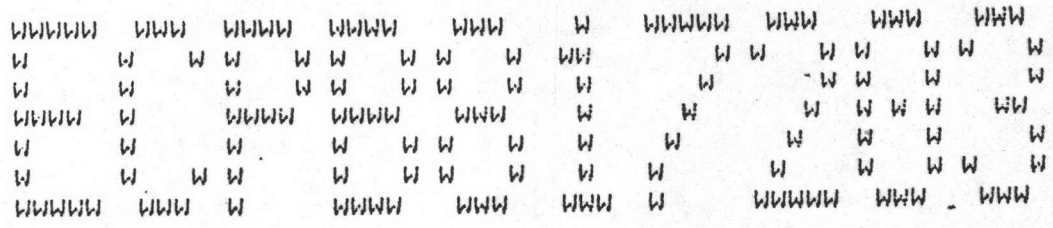
```
+ 0,   170,    9,3,     0,   50,2,   100,    0,3,   -50,  -30,
+ 2,     0,   60,3,   -35,    0,2,    70,  -60,3,   -70,    0,
+ 2,    70,   60,
#    /*  *** + *** */
+ 0,   171,    5,3,    50,   20,2,     0,   60,3,   -30,  -30,
+ 2,    60,    0,
#    /*  *** , *** */
+ 0,   172,    8,3,    60,   10,2,   -20,    0,2,     0,   10,
+ 2,    20,    0,2,     0,  -10,2,   -20,  -10,2,    10,   10,
#    /*  *** - *** */
+ 0,   173,    3,3,    30,   50,2,    60,    0,
#    /*  *** . *** */
+ 0,   174,    6,3,    60,    0,2,   -20,    0,2,     0,   10,
+ 2,    20,    0,2,     0,  -10,
#    /*  *** / *** */
+ 0,   175,    2,2,   100,  100,
#    /*  *** : *** */
+ 0,   186,   11,3,    40,   20,2,    20,    0,2,     0,   10,
+ 2,   -20,    0,2,     0,  -10,3,     0,   50,2,    20,    0,
+ 2,     0,   10,2,   -20,    0,2,     0,  -10,
#    /*  *** ; *** */
+ 0,   187,   13,3,    60,   20,2,   -20,    0,2,     0,   10,
+ 2,    20,    0,2,     0,  -10,2,   -20,  -10,2,    10,   10,
+ 3,   -10,   50,2,    20,    0,2,     0,   10,2,   -20,    0,
+ 2,     0,  -10,
#    /*  *** < *** */
+ 0,   188,    4,3,   100,   80,2,  -100,  -30,2,   100,  -30,
#    /*  *** = *** */
+ 0,   189,    5,3,     0,   40,2,   100,    0,3,     0,   20,
+ 2,  -100,    0,
#    /*  *** > *** */
+ 0,   190,    4,3,     0,   20,2,   100,   30,2,  -100,   30,
#    /*  *** ? *** */
+ 0,   191,   15,3,     0,   80,2,    20,   20,2,    60,    0,
+ 2,    20,  -20,2,     0,  -10,2,   -20,  -20,2,   -30,    0,
+ 2,   -20,  -20,2,     0,  -10,3,   -10,  -20,2,    20,    0,
+ 2,     0,   10,2,   -20,    0,2,     0,  -10,
#    /* *** @ *** */
+ 0,   192,   21,3,    80,   70,2,   -10,   10,2,   -40,    0,
+ 2,   -10,  -10,2,     0,  -40,2,    10,  -10,2,    40,    0,
+ 2,    10,   10,2,     0,   40,3,    -5,  -45,2,    25,    0,
+ 3,   -80,  -25,2,    60,    0,2,    20,   20,2,     0,   60,
+ 2,   -20,   20,2,   -60,    0,2,   -20,  -20,2,     0,  -60,
+ 2,    20,  -20,
#    /*  *** [ *** */
+ 0,   219,    5,3,   100,    0,2,   -60,    0,2,     0,  100,
+ 2,    60,    0,
#    /*  *** ] *** */
+ 0,   221,    4,2,    60,    0,2,     0,  100,2,   -60,    0,
#    /*  *** \ *** */
+ 0,   220,    3,3,   100,    0,2,  -100,  100,
#    /*  *** ^ *** */
+ 0,   222,    6,3,    50,    0,2,     0,  100,2,   -30,  -40,
+ 3,    60,    0,2,   -30,   40,
#    /*  *** _ *** */
+ 0,   223,    6,3,     0,   50,2,   100,    0,3,   -50,   20,
+ 2,   -50,  -20,2,    50,  -20,
#    /* *** a *** */
+ 0,   225,   13,3,    20,    0,2,    50,    0,2,    20,   20,
+ 2,     0,   20,2,   -20,   20,2,   -50,    0,2,   -20,  -20,
```

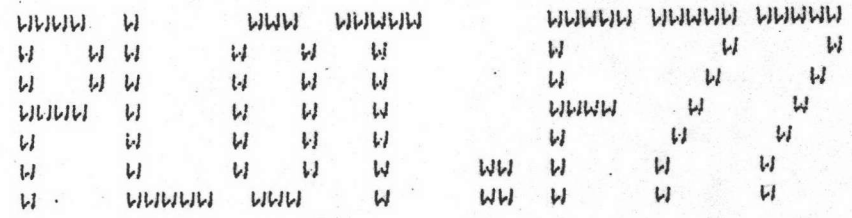
```
C
     + 2,     0,  -20, 2,    20,  -20, 3,    70,   20, 2,     0,  -10,
     + 2,    10,  -10,
     #    /* *** h *** */
     + 0,   226,   10, 2,     0,  100, 3,     0,  -60, 2,    20,   20,
     + 2,    60,    0, 2,    20,  -20, 2,     0,  -20, 2,   -20,  -20,
     + 2,   -60,    0, 2,   -20,   20,
     #    /* *** c *** */
     + 0,   227,    9, 3,   100,   40, 2,   -20,   20, 2,   -60,    0,
     + 2,   -20,  -20, 2,     0,  -20, 2,    20,  -20, 2,    60,    0,
     + 2,    20,   20,
     #    /* *** d *** */
     + 0,   228,   11, 3,   100,    0, 2,     0,  100, 3,     0,  -60,
     + 2,   -20,   20, 2,   -60,    0, 2,   -20,  -20, 2,     0,  -20,
     + 2,    20,  -20, 2,    60,    0, 2,    20,   20,
     #    /* *** e *** */
     + 0,   229,   11, 3,     0,   30, 2,   100,    0, 2,     0,   10,
     + 2,   -20,   20, 2,   -60,    0, 2,   -20,  -20, 2,     0,  -20,
     + 2,    20,  -20, 2,    60,    0, 2,    20,   10,
     #    /* *** f *** */
     + 0,   230,    8, 3,   100,   80, 2,   -20,   20, 2,   -50,    0,
     + 2,   -20,  -20, 2,     0,  -80, 3,   -10,   50, 2,    50,    0,
     #    /* *** g *** */
     + 0,   231,   15, 3,    20,    0, 2,    60,    0, 2,    20,   20,
     + 2,     0,   20, 2,   -20,   20, 2,   -60,    0, 2,   -20,  -20,
     + 2,     0,  -20, 2,    20,  -20, 3,    80,   20, 2,     0,  -50,
     + 2,   -20,  -20, 2,   -60,    0, 2,   -20,   20,
     #    /* *** h *** */
     + 0,   232,    7, 2,     0,  100, 3,     0,  -60, 2,    20,   20,
     + 2,    60,    0, 2,    20,  -20, 2,     0,  -40,
     #    /* *** i *** */
     + 0,   233,    8, 3,    50,    0, 2,     0,   60, 3,    -3,   20,
     + 2,     6,    0, 2,     0,   -4, 2,    -6,    0, 2,     0,    4,
     #    /* *** j *** */
     + 0,   234,   11, 3,    95,   60, 2,     0,  -90, 2,   -20,  -20,
     + 2,   -55,    0, 2,   -20,   20, 3,    92,  110, 2,     6,    0,
     + 2,     0,   -4, 2,    -6,    0, 2,     0,    4,
     #    /* *** k *** */
     + 0,   235,    6, 2,     0,  100, 3,     0,  -60, 2,    80,   30,
     + 3,   -80,  -30, 2,   100,  -40,
     #    /* *** l *** */
     + 0,   236,    3, 3,    50,    0, 2,     0,  100,
     #    /* *** m *** */
     + 0,   237,   11, 2,     0,   40, 2,    20,   20, 2,    10,    0,
     + 2,    20,  -20, 2,     0,  -20, 3,     0,   20, 2,    20,   20,
     + 2,    10,    0, 2,    20,  -20, 2,     0,  -40,
     #    /* *** n *** */
     + 0,   238,    7, 2,     0,   60, 3,     0,  -20, 2,    20,   20,
     + 2,    60,    0, 2,    20,  -20, 2,     0,  -40,
     #    /* *** o *** */
     + 0,   239,   10, 3,    20,    0, 2,    60,    0, 2,    20,   20,
     + 2,     0,   20, 2,   -20,   20, 2,   -60,    0, 2,   -20,  -20,
     + 2,     0,  -20, 2,    20,  -20,
     #    /* *** p *** */
     + 0,   240,   11, 3,     0,  -50, 2,     0,  110, 3,     0,  -20,
     + 2,    20,   20, 2,    60,    0, 2,    20,  -20, 2,     0,  -20,
     + 2,   -20,  -20, 2,   -60,    0, 2,   -20,   20,
     #    /* *** q *** */
     + 0,   241,   11, 3,   100,  -50, 2,     0,  110, 3,     0,  -20,
     + 2,   -20,   20, 2,   -60,    0, 2,   -20,  -20, 2,     0,  -20,
     + 2,    20,  -20, 2,    60,    0, 2,    20,   20,
```

```
#    /* *** r *** */
+ 0,   242,    6, 2,     0,   60, 3,     0,  -20, 2,    20,   20,
+ 2,    60,    0, 2,    20,  -20,
#    /* *** s *** */
+ 0,   243,   13, 3,     0,   10, 2,    10,  -10, 2,    80,    0,
+ 2,    10,   10, 2,     0,   10, 2,   -10,   10, 2,   -80,    0,
+ 2,   -10,   10, 2,     0,   10, 2,    10,   10, 2,    80,    0,
+ 2,    10,  -10,
#    /* *** t *** */
+ 0,   244,    5, 3,    50,    0, 2,     0,  100, 3,   -40,  -30,
+ 2,    80,    0,
#    /* *** u *** */
+ 0,   245,    7, 3,     0,   60, 2,     0,  -40, 2,    20,  -20,
+ 2,    60,    0, 2,    20,   20, 2,     0,   40,
#    /* *** v *** */
+ 0,   246,    4, 3,     0,   60, 2,    50,  -60, 2,    50,   60,
#    /* *** w *** */
+ 0,   247,   12, 3,     0,   60, 2,     0,  -40, 2,    20,  -20,
+ 2,    10,    0, 2,    20,   20, 2,     0,   10, 3,     0,  -10,
+ 2,    20,  -20, 2,    10,    0, 2,    20,   20, 2,     0,   40,
#    /* *** x *** */
+ 0,   248,    5, 3,     0,   60, 2,   100,  -60, 3,  -100,    0,
+ 2,   100,   60,
#    /* *** y *** */
+ 0,   249,   12, 3,     0,   60, 2,     0,  -40, 2,    20,  -20,
+ 2,    60,    0, 2,    20,   20, 2,     0,   40, 3,     0,  -40,
+ 2,     0,  -50, 2,   -20,  -20, 2,   -60,    0, 2,   -20,   20,
#    /* *** z *** */
+ 0,   250,    5, 3,     0,   60, 2,   100,    0, 2,  -100,  -60,
+ 2,   100,    0,
#    /*  *** SYMBOL 0 *** */
+ 0,   000,    5, 2,   100,    0, 2,     0,  100, 2,  -100,    0,
+ 2,     0, -100,
#    /*  *** SYMBOL 1 *** */
+ 0,   001,    4, 2,    50,  100, 2,    50, -100, 2,  -100,    0,
#    /*  *** SYMBOL 2 *** */
+ 0,   002,   10, 3,     0,   20, 2,    20,  -20, 2,   -60,    0,
+ 2,    20,   20, 2,     0,   60, 2,   -20,   20, 2,   -60,    0,
+ 2,   -20,  -20, 2,     0,  -60,
#    /*  *** SYMBOL 3 *** */
+ 0,   003,    6, 3,    50,    0, 2,    50,   50, 2,   -50,   50,
+ 2,   -50,  -50, 2,    50,  -50,
#    /*  *** SYMBOL 4 *** */
+ 0,   004,    5, 3,    50,    0, 2,     0,  100, 3,   -50,  -50,
+ 2,   100,    0,
#    /*  *** SYMBOL 5 *** */
+ 0,   005,    8, 2,   100,  100, 3,  -100,    0, 2,   100, -100,
+ 3,   -50,    0, 2,     0,  100, 3,   -50,  -50, 2,   100,    0,
#    /*  *** SYMBOL 6 *** */
+ 0,   006,    4, 2,   100,  100, 3,  -100,    0, 2,   100, -100,
#    /*  *** SYMBOL 7 *** */
+ 0,   007,    8, 2,    25,   25, 2,     0,   50, 2,    50,    0,
+ 2,     0,  -50, 2,   -50,    0, 3,    25,   25, 2,    50,   50,
#    /*  *** SYMBOL 8 *** */
+ 0,   008,    6, 3,    50,    0, 2,     0,  100, 2,   -50,  -50,
+ 2,   100,    0, 2,   -50,   50,
#    /*  *** SYMBOL 9 *** */
+ 0,   009,    5, 2,   100,  100, 2,  -100,    0, 2,   100, -100,
+ 2,  -100,    0,
# -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,3570*-1,-1,-1/
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>PLOT.F77





Label: PRT014 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>PLOT.F77
File last modified: 87-09-03. 15:04:04. Thu

Spooled: 87-09-18. 10:37:48. Fri [Spooler rev 19.4]
Started: 87-09-18. 10:53:36. Fri on: PRO by: PRO

```
C
C
C
C
C
      SUBROUTINE  AXIS3D(TH)
      CALL  PLOT3D(0.0,0.0,0.0,TH,3)
      CALL  PLOT3D(4.1,0.0,0.0,TH,2)
      CALL  PLOT3D(0.0,0.0,0.0,TH,3)
      CALL  PLOT3D(0.0,4.1,0.0,TH,2)
      CALL  PLOT3D(0.0,0.0,0.0,TH,3)
      CALL  PLOT3D(0.0,0.0,4.1,TH,2)
      CALL  PLOT3D(0.0,0.0,0.0,TH,3)
      RETURN
      END

C
C
C
      SUBROUTINE PLOT3D(X,Y,Z,TH,IPEN)

      CALL PLOT(Y-X*(COSD(TH)),Z-X*(SIND(TH)),IPEN)
C      write(15,*)'plot3d',y-x*cosd(th),z-x*sind(th),ipen

      RETURN
      END

C
C     GENERATE RELATIVE PLOT WITH ROTATION
C
      SUBROUTINE RELPLT(RELX,RELY,IPEN)

      INTEGER*4   IPEN
      REAL*4      RELX,RELY
C
C     PARAMETERS :
C                 RELX,RELY      DEFINE RELATIVE COORDS FOR END OF VECTOR
C                 IPEN           IS THE DRAW/MOVE OPTION (SEE 'PLOT')
C               * TCOS           IS THE COSINE OF THE ANGLE OF ROTATION
C               * TSIN           IS THE SIN OF THE ANGLE OF ROTATION
C
C     NOTE:   * DENOTES THESE  VALUES ARE IN COMMON
C

$INSERT LGDEVICE.F77

      REAL*4 ROTATX,ROTATY

      ROTATX=((RELX*TCOS)-(RELY*TSIN))
      ROTATY=((RELX*TSIN)+(RELY*TCOS))
      CALL RPLOT(ROTATX,ROTATY,IPEN)          /* PLOT THE VECTOR

      RETURN
      END


C
C     RELATIVE PEN MOVEMENT CONTROL
C
      SUBROUTINE RPLOT(X,Y,IPEN)
```

```
C
      INTEGER*4   IPEN
      REAL*4      X,Y
C
C     PARAMETERS :
C                 X,Y   ARE RELATIVE COORDINATES TO THE CURRENT POSITION
C                 IPEN  IS THE 'PLOT' IPEN OPTION
C                       SEE SUBROUTINE PLOT
C
$INSERT LGDEVICE.F77

      REAL*4 CX,CY,FDUMMY

      CALL WHERE(CX,CY,FDUMMY)
      CALL PLOT(X+CX,Y+CY,IPEN)
      RETURN
      END

C
C     BASIC 'PEN' MOVEMENT CONTROL SUBROUTINE
C
      SUBROUTINE PLOT(XS,YS,IPEN)

      INTEGER*4   IPEN
      REAL*4      XS,YS
C
C     PARAMETERS :
C                 X,Y   SPECIFY THE (REAL) COORDINATES TO WHICH THE PEN WILL
C                       BE MOVED
C                 IPEN  IS THE PEN MOVE OPTION, ONE OF THE BELOW:
C                       2  PEN DOWN (DRAW)
C                       3  PEN UP   (MOVE)
C                      -2  PEN DOWN, RESET ORIGIN TO X,Y
C                      -3  PEN UP, RESET ORIGIN TO X,Y
C
$INSERT LGDEVICE.F77
$INSERT HEADFILE.F77

      INTEGER*2 ISX,ISY,LDEV,BUFFX,BUFFY
      REAL*4    X,Y,XT,YT,FT

c      write(15,*)'plot',x,y,ipen

      X = XS
      Y = YS
      CALL WHERE(XT,YT,FT)
      IF (YS.EQ.999.) Y = YT
      IF (XS.EQ.999.) X = XT

      IF (IABS(IPEN).EQ.999) GOTO 9000

      IF(IABS(IPEN).NE.2.AND.IABS(IPEN).NE.3)GO TO 75 /* BAD CALL
      IF(X.LT.0.)GO TO 1
      ISX=IFIX(X*SFACTR*RPIX+.5)+XORG        /* ABSOLUTE X COORDINATE
      GO TO 2
1     ISX=IFIX(X*SFACTR*RPIX-.5)+XORG
2     IF(Y.LT.0.)GO TO 3
      ISY=IFIX(Y*SFACTR*RPIY+.5)+YORG        /* ABSOLUTE Y COORDINATE
      GO TO 4
3     ISY=IFIX(Y*SFACTR*RPIY-.5)+YORG
```

```
C
4       IF(ISX.LT.0.OR.ISY.LT.0)GO TO 50        /* GENERAL OUT OF RANGE CHECKS
        IF (VERTPLOT) GOTO 20

        CONTINUE
        IF (ISX.GT.HEADXMAX) GOTO 50
        IF (ISY.GT.HEADYMAX) GOTO 50
        GOTO 30

20      CONTINUE
        IF (ISX.GT.HEADYMAX) GOTO 50
        IF (ISY.GT.HEADXMAX) GOTO 50
        GOTO 30

30      CONTINUE
        IF ((MIRROR.LT.INTS(0)).OR.(MIRROR.GT.INTS(3))) MIRROR=INTS(0)
        IF (MIRROR.EQ.1) GOTO 1000
        IF (MIRROR.EQ.2) GOTO 2000
        IF (MIRROR.EQ.3) GOTO 3000
        GOTO 5000

1000    CONTINUE
        ISX = HEADXMAX-ISX
        GOTO 5000

2000    CONTINUE
        ISY = HEADYMAX-ISY
        GOTO 5000

3000    CONTINUE
        ISX = HEADXMAX-ISX
        ISY = HEADYMAX-ISY
        GOTO 5000

5000    CONTINUE

C
C       SAVE LAST VALID X & Y

        LASTX=X
        LASTY=Y

C
C       ELIMINATE LAST POINT IF IT WAS A MOVE AND THIS IS ONE TOO

        IF(( OUTBUFPT.GT.1 ).AND.( IABS(IPEN).EQ.3 ).AND.
       X    ( OUTBUF(2,OUTBUFPT-1).LT.0) ) OUTBUFPT=OUTBUFPT-1

        IF(OUTBUFPT.GT.OUTBUFSZ)CALL CLRBUF      /* CLEAR OUTPUT BUFFER

C
C       CHECK FOR HORIZONTAL PLOT, FOR WHICH X AND Y GET "SWITCHED".

        BUFFX=ISX    /* X COORD FOR BUFFER
        BUFFY=ISY    /* Y COORD
        IF (.NOT.VERTPLOT) GOTO 40   /* ALL SET IF VERTICAL PLOT
                                     /* USE LDEV AS TEMP FOR A SEC
          LDEV=BUFFY
          BUFFY=HEADYMAX-ISX
          BUFFX=LDEV

C
```

```
C
C       IF OUTPUTTING TO FILE, SAVE MAXIMUM COORDINATES IN HEADER
C
40      IF (.NOT.FILEPLOT) GOTO 45                  /* NOT GOING TO FILE
          IF (BUFFX.GT.HEADXHI) HEADXHI=BUFFX       /* SAVE.X IF GREATER THAN LAST
          IF (BUFFY.GT.HEADYHI) HEADYHI=BUFFY
C
45      CONTINUE
        BUFFY = BUFFY + INTS(1)         /* -0 = 0 SO ADD 1
        IF (IABS(IPEN).EQ.3) BUFFY=-BUFFY                   /* MOVE/DRAW INDICATOR
C
C       INSERT POINTS INTO BUFFER
C
c        write(15,*)'plot outbuffpt',buffx,buffy,OUTBUFPT
        OUTBUF(1,OUTBUFPT)=BUFFX        /* COORDINATE X
        OUTBUF(2,OUTBUFPT)=BUFFY        /* COORDINATE Y,M/D,LNWT
c        write(15,*)'plot',OUTBUF(1,OUTBUFPT),
c       X               and(OUTBUF(2,OUTBUFPT),ints(:007777))
C
C       SET UP NEW CURRENT POSITION
C
        CURX=ISX
        CURY=ISY
C
C       SET NEW ORIGIN IF REQUESTED
C
        IF(IPEN.GT.0)GO TO 5
        XORG=ISX
        YORG=ISY
        LASTX=0.                                    /* RESET FOR ORIGIN-RELATIVE
        LASTY=0.
C
5       OUTBUFPT=OUTBUFPT+1                         /* INCREMENT OUTPUT BUFFER POINTER
        GOTO 9999
C
C     HANDLE COORDINATE-OUT-OF-RANGE ERROR
C
50        CONTINUE
c          write(15,*)'plot error 50'
          WRITE(ERRDEV,51)X,Y,IPEN,SFACTR,XORG,YORG,CURX,CURY,ISX,ISY
51        FORMAT(' PLOT RANGE ERROR:'/
     1          5X,'SPECIFIED COORDINATES = (',F9.3,1H,,F9.3,1H),
     2          3X,'(IPEN=',I1,1H)/
     3          5X,'SCALE FACTOR = ',F10.3/
     4          5X,'ABSOLUTE ORIGIN AT (',I6,1H,,I6,1H),'  (RASTERS)'/
     5          5X,'CURRENT COORDINATES = (',I6,1H,,I6,')  (RASTERS)'/
     6          5X,'RESULTANT COORDINATES = (',I6,1H,,I6,1H),'  (RASTERS)
     7          /)
C
          GOTO 9999
C
C         HANDLER FOR BAD CALL
C
75        CONTINUE
c          write(15,*)'plot error 75',errdev
C          WRITE(ERRDEV,80)IPEN
```

```
C

80      FORMAT(/' BAD CALL TO PLOT - IPEN=',I6/)

9000    CONTINUE
C       write(15,*)'plot 999'
        CALL ENDPLT
        RETURN

9999    CONTINUE
        IF (INTERACT) CALL CLRBUF
        RETURN
        END


C
C       CIRCLE OF CALCOMP
C
        SUBROUTINE CIRCLE(P1,P2,P3,P4,P5,IMODE)

        REAL*4      P1,P2,P3,P4,P5
        INTEGER*4   IMODE

        REAL*4      ANGSTART,ANGEND,R,K,A,B,C,M,TEMP1,TEMP2,TEMP3
        REAL*4      XS,YS,XE,YE,XC,YC,AS,AE
        INTEGER*4   IMODE2,IMODE3

        IMODE2 = IABS(IMODE)
        IF ((IMODE2.LT.1).AND.(IMODE2.GT.6)) IMODE2 =7
        IMODE3 = IMODE2
        IF (IMODE.LT.0) IMODE3 = -IMODE2
        GOTO (1000,1400,1200,1600,1800,2000,2200)IMODE2

1000    CONTINUE
        XS = P1
        YS = P2
        XC = P3
        YC = P4
        ANGSTART = ATAN2(YS-YC,XS-XC)*57.29578
        ANGEND   = ANGSTART+359.999
        IF (IMODE.LT.0) ANGEND = ANGSTART+.002
        IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
        IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.
        R        = ( (ABS((YS-YC))**2) + (ABS((XS-XC))**2) )**.5
        GOTO 9000

1200    CONTINUE
        XS = P1
        YS = P2
        XE = P3
        YE = P4
        R  = P5
        IF (XS.EQ.XE) GOTO 1270
        IF (YS.EQ.YE) GOTO 1280

        M  = (XE-XS)/(YS-YE)
        K  = (ABS(XS)**2.+ABS(YS)**2.-ABS(XE)**2.-ABS(YE)**2.)
     X     /(2.*(YS-YE))

        A  = 1.+(ABS(M)**2.)
        B  = -(2.*(XE-(M*K)+(YE*M)))
```

```
C
      C   = (ABS(XE)**2.)+(ABS(YE)**2.)
     X     -(ABS(R)**2.)-(2.*YE*K)+(ABS(K)**2.)

      TEMP1 = (-B)/(2.*A)
      TEMP2 = (ABS(B)**2.)-(4.*A*C)
      IF (TEMP2.LT.0.) TEMP2 = 0.
      TEMP2 = TEMP2**.5/(2.*A)

1250  XC    = TEMP1+TEMP2
      YC    = M*XC + K
      ANGSTART = ATAN2(YS-YC,XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC,XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.

      IF ( (ANGEND.GE.ANGSTART)          .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGEND-ANGSTART).GT.180.) .AND.
     X    (IMODE.GE.0) ) GOTO 1299
       IF ( (ANGEND.LT.ANGSTART)             .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGSTART-ANGEND).LT.180.)      .AND.
     X    (IMODE.GE.0) ) GOTO 1299

       IF ( (ANGSTART.GE.ANGEND)             .AND.      /* CLOCKWISE
     X    ((ANGSTART-ANGEND).GT.180.)      .AND.
     X    (IMODE.LT.0) ) GOTO 1299
       IF ( (ANGSTART.LT.ANGEND)             .AND.      /* CLOCKWISE
     X    ((ANGEND-ANGSTART).LT.180.)      .AND.
     X    (IMODE.LT.0) ) GOTO 1299

      TEMP2 = -(TEMP2)
      XC    = TEMP1+TEMP2
      YC    = M*XC + K
      ANGSTART = ATAN2(YS-YC,XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC,XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.
      GOTO 1299
1270  YC = (YE+YS)/2.
      TEMP1 = XS
      TEMP2 = ((ABS(R)**2.)-(ABS(YC-YS)**2.))**.5
      XC = TEMP1 +TEMP2
      ANGSTART = ATAN2(YS-YC,XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC,XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.

       IF ( (ANGEND.GE.ANGSTART)          .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGEND-ANGSTART).GT.180.) .AND.
     X    (IMODE.GE.0) ) GOTO 1299
       IF ( (ANGEND.LT.ANGSTART)             .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGSTART-ANGEND).LT.180.)      .AND.
     X    (JMODE.GE.0) ) GOTO 1299

       IF ( (ANGSTART.GE.ANGEND)             .AND.      /* CLOCKWISE
     X    ((ANGSTART-ANGEND).GT.180.)      .AND.
     X    (IMODE.LT.0) ) GOTO 1299
       IF ( (ANGSTART.LT.ANGEND)             .AND.      /* CLOCKWISE
     X    ((ANGEND-ANGSTART).LT.180.)      .AND.
     X    (IMODE.LT.0) ) GOTO 1299
```

```
C
      TEMP2 = -(TEMP2)
      XC    = TEMP1+TEMP2
      ANGSTART = ATAN2(YS-YC, XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC, XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.
      GOTO 1299

 1200 XC = (XE+XS)/2.
      TEMP1 = YS
      TEMP2 = ((ABS(R)**2.)-(ABS(XC-XS)**2.))**.5
      YC = TEMP1 +TEMP2
      ANGSTART = ATAN2(YS-YC, XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC, XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.

      IF ( (ANGEND.GE.ANGSTART)          .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGEND-ANGSTART).GT.180.) .AND.
     X    (IMODE.GE.0) ) GOTO 1299
      IF ( (ANGEND.LT.ANGSTART)          .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGSTART-ANGEND).LT.180.)    .AND.
     X    (IMODE.GE.0) ) GOTO 1299

      IF ( (ANGSTART.GE.ANGEND)          .AND.      /* CLOCKWISE
     X    ((ANGSTART-ANGEND).GT.180.)    .AND.
     X    (IMODE.LT.0) ) GOTO 1299
      IF ( (ANGSTART.LT.ANGEND)          .AND.      /* CLOCKWISE
     X    ((ANGEND-ANGSTART).LT.180.)    .AND.
     X    (IMODE.LT.0) ) GOTO 1299

      TEMP2 = -(TEMP2)
      YC    = TEMP1+TEMP2
      ANGSTART = ATAN2(YS-YC, XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC, XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.
      GOTO 1299

 1299 CONTINUE
      GOTO 9000

 1400 CONTINUE
      XS = P1
      YS = P2
      XE = P3
      YE = P4
      R  = P5

      IF (XS.EQ.XE) GOTO 1470
      IF (YS.EQ.YE) GOTO 1490

      M  = (XE-XS)/(YS-YE)
      K  = (ABS(XS)**2.+ABS(YS)**2.-ABS(XE)**2.-ABS(YE)**2.)
     X     /(2.*(YS-YE))
C     write(15,*)'circle pass *'

      A  = 1.+(ABS(M)**2.)
      B  = -(2.*(XE-(M*K)+(YE*M)))
      C  = (ABS(XE)**2.)+(ABS(YE)**2.)
```

```
C:

     X     -(ABS(R)**2. )-(2. *YE*K)+(ABS(K)**2. )
C       write(15,*)'circle', M, K, a, b, c

      TEMP1 = (-B)/(2. *A)
      TEMP2 = (ABS(B)**2. )-(4. *A*C)
      IF (TEMP2. LT. 0 ) TEMP2 = 0.
      TEMP2 = TEMP2**. 5/(2. *A)

1450  XC     = TEMP1+TEMP2
      YC     = M*XC + K
      ANGSTART = ATAN2(YS-YC, XS-XC)*57. 29578
      ANGEND   = ATAN2(YE-YC, XE-XC)*57. 29578
      IF (ANGSTART. LT. 0. ) ANGSTART = ANGSTART + 360.
      IF (ANGEND. LT. 0. ) ANGEND = ANGEND + 360.

      IF ( (ANGEND. GE. ANGSTART)              . AND.      /* COUNTER-CLOCKWISE
     X    ((ANGEND-ANGSTART). GT. 180. )       . AND.
     X    (IMODE. GE. 0) ) GOTO 1460
      IF ( (ANGEND. LT. ANGSTART)              . AND.      /* COUNTER-CLOCKWISE
     X    ((ANGSTART-ANGEND). LT. 180. )       . AND.
     X    (IMODE. GE. 0) ) GOTO 1460
C       write(15,*)'circle pass $'

      IF ( (ANGSTART. GE. ANGEND)              . AND.       /* CLOCKWISE
     X    ((ANGSTART-ANGEND). GT. 180. )       . AND.
     X    (IMODE. LT. 0) ) GOTO 1460
      IF ( (ANGSTART. LT. ANGEND)              . AND.       /* CLOCKWISE
     X    ((ANGEND-ANGSTART). LT. 180. )       . AND.
     X    (IMODE. LT. 0) ) GOTO 1460
C       write(15,*)'circle pass $$$$$'
      GOTO 1499

1460  CONTINUE                                          /* CHANGE CENTER POINT
      TEMP2 = -(TEMP2)
      XC     = TEMP1+TEMP2
      YC     = M*XC+K
      ANGSTART = ATAN2(YS-YC, XS-XC)*57. 29578
      ANGEND   = ATAN2(YE-YC, XE-XC)*57. 29578
      IF (ANGSTART. LT. 0. ) ANGSTART = ANGSTART + 360.
      IF (ANGEND. LT. 0. ) ANGEND = ANGEND + 360.
C       write(15,*)'circle pass === ', XC, YC, ANGSTART, ANGEND
      GOTO 1499

1470  YC = (YE+YS)/2.
      TEMP1 = XS
      TEMP2 = ((ABS(R)**2. )-(ABS(YC-YS)**2. ))**. 5
      XC = TEMP1 +TEMP2
      ANGSTART = ATAN2(YS-YC, XS-XC)*57. 29578
      ANGEND   = ATAN2(YE-YC, XE-XC)*57. 29578
      IF (ANGSTART. LT. 0. ) ANGSTART = ANGSTART + 360.
      IF (ANGEND. LT. 0. ) ANGEND = ANGEND + 360.

      IF ( (ANGEND. GE. ANGSTART)              . AND.      /* COUNTER-CLOCKWISE
     X    ((ANGEND)-ANGSTART). GT. 180. )      . AND.
     X    (IMODE. GE. 0) ) GOTO 1475
      IF ( (ANGEND. LT. ANGSTART)              . AND.      /* COUNTER-CLOCKWISE
     X    ((ANGSTART-ANGEND). LT. 180. )       . AND.
     X    (IMODE. GE. 0) ) GOTO 1475
```

```
C
      IF ( (ANGSTART.GE.ANGEND)                    .AND.      /* CLOCKWISE
     X    ((ANGSTART-ANGEND).GT.180.)              .AND.
     X    (IMODE.LT.0) ) GOTO 1475
      IF ( (ANGSTART.LT.ANGEND)                    .AND.      /* CLOCKWISE
     X    ((ANGEND-ANGSTART).LT.180.)              .AND.
     X    (IMODE.LT.0) ) GOTO 1475
      GOTO 1499

1475  TEMP2 = -(TEMP2)
      XC    = TEMP1+TEMP2
      ANGSTART = ATAN2(YS-YC,XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC,XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.
      GOTO 1499

1490  XC = (XE+XS)/2.
      TEMP1 = YS
      TEMP2 = ((ABS(R)**2.)-(ABS(XC-XS)**2.))**.5
      YC = TEMP1 +TEMP2
      ANGSTART = ATAN2(YS-YC,XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC,XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.

      IF ( (ANGEND.GE.ANGSTART)                    .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGEND-ANGSTART).GT.180.)              .AND.
     X    (IMODE.GE.0) ) GOTO 1495
      IF ( (ANGEND.LT.ANGSTART)                    .AND.      /* COUNTER-CLOCKWISE
     X    ((ANGSTART-ANGEND).LT.180.)              .AND.
     X    (IMODE.GE.0) ) GOTO 1495

      IF ( (ANGSTART.GE.ANGEND)                    .AND.      /* CLOCKWISE
     X    ((ANGSTART-ANGEND).GT.180.)              .AND.
     X    (IMODE.LT.0) ) GOTO 1495
      IF ( (ANGSTART.LT.ANGEND)                    .AND.      /* CLOCKWISE
     X    ((ANGEND-ANGSTART).LT.180.)              .AND.
     X    (IMODE.LT.0) ) GOTO 1495
      GOTO 1499

1495  TEMP2 = -(TEMP2)
      YC    = TEMP1+TEMP2
      ANGSTART = ATAN2(YS-YC,XS-XC)*57.29578
      ANGEND   = ATAN2(YE-YC,XE-XC)*57.29578
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART + 360.
      IF (ANGEND.LT.0.) ANGEND = ANGEND + 360.
      GOTO 1499

1499  GOTO 9000

1600  CONTINUE
      XS = P1
      YS = P2
      ANGSTART = P3
      ANGSTART = AMOD(ANGSTART,360.)
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART +360.
      ANGEND   = ANGSTART+359.999
      ANGEND   = AMOD(ANGEND,360.)
      IF (ANGEND.LT.0.) ANGEND = ANGEND +360.
      IF (IMODE.LT.0) ANGEND = ANGSTART+.002
```

```
C
      R          = P5
      XC     = XS - COSD(ANGSTART)
      YC     = YS - SIND(ANGSTART)
      GOTO 9000

1800  CONTINUE
      XS = P1
      YS = P2
      ANGSTART = P3
      ANGSTART = AMOD(ANGSTART, 360. )
      IF (ANGSTART. LT. 0. ) ANGSTART = ANGSTART +360.
      ANGEND   = P4
      ANGEND   = AMOD(ANGEND, 360. )
      IF (ANGEND. LT. 0. ) ANGEND = ANGEND +360.
      R  = P5
      XC     = XS - COSD(ANGSTART).
      YC     = YS - SIND(ANGSTART)
      GOTO 9000

2000  CONTINUE
      XC         = P1
      YC         = P2
      ANGSTART = P3
      ANGSTART = AMOD(ANGSTART, 360. )
      IF (ANGSTART. LT. 0. ) ANGSTART = ANGSTART +360.
      ANGEND   = ANGSTART+359. 999
      ANGEND   = AMOD(ANGEND, 360. )
      IF (ANGEND. LT. 0. ) ANGEND = ANGEND +360.
      IF (IMODE. LT. 0) ANGEND = ANGSTART+. 002
      R          = P5
      GOTO 9000

2200  CONTINUE
      XC         = P1
      YC         = P2
      ANGSTART = P3.
      ANGSTART = AMOD(ANGSTART, 360. )
      IF (ANGSTART. LT. 0. ) ANGSTART = ANGSTART +360.
      ANGEND   = P4
      ANGEND   = AMOD(ANGEND, 360. )
      IF (ANGEND. LT. 0. ) ANGEND = ANGEND +360.
      R          = P5
      GOTO 9000

9000  CONTINUE
c      write(15,*)'circle', angstart, angend
      CALL XCIRCLE(XC, YC, R, ANGSTART, ANGEND, IMODE3)
      RETURN
      END
C
C     CREATE CIRCLE ROUTINE
C
      SUBROUTINE XCIRCLE(XC, YC, R, AS, AE, IMODE)

      REAL*4      XC, YC, R, AS, AE
      INTEGER*4   IMODE

C
C     PARAMETERS :
C                 XC
```

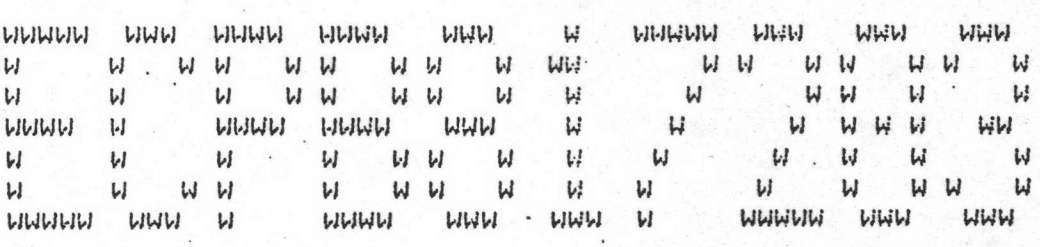
```
C
C                 YC        CENTER POINT
C                 R         RADIUS
C      LOCAL VARS :
C                 XPLOT   ARRAY OF POINT OF CIRCLE ARC
C                 YPLOT
C
*INSERT LGDEVICE.F77

      REAL*4      XPLOT,YPLOT
      REAL*4      ANGINC,ANGLE
      REAL*4      TEMP1,TEMP2,TEMP3,TEMP4
      REAL*4      XSTART,YSTART,ANGSTART,ANGEND
      INTEGER*4   NUMPLOT,I,J,IQUAD,ISTART,IEND,LOOP,INCREASE
      INTEGER*4   ITEMP1,ITEMP2,ITEMP3,ITEMP4,IQ1,IQ2,IQ3,IQ4
      COMMON      /CIRPLOT/XPLOT(12880),YPLOT(12880)

      ANGSTART = AMOD(AS,360.)
      ANGEND   = AMOD(AE,360.)
      IF (ANGSTART.LT.0.) ANGSTART = ANGSTART +360.
      IF (ANGEND.LT.0.)   ANGEND = ANGEND +360.

c      write(15,*)xc,yc,r,'xcircle',angstart,angend,imode

      NUMPLOT = IFIX( ((SFACTR*RPIX*R)*11.)/7. ) /* 11/7 = 2*(22/7)*radius/4

      IF (NUMPLOT.GT.3220) NUMPLOT = 3220  /* SCREEN HAVE MAX 1024 LENGTH
      ANGINC = 0.
      IF (NUMPLOT.EQ.0) GOTO 100
      ANGINC = 90./REAL(NUMPLOT)           /* MAX RADIUS = 1024/2 = 512
                                           /* MAX CIRCLE = 2*(22/7)*512
100   ANGLE  = 0.                          /* 1/4 MAX CIR = (11/7)*512 = 805

      IQ2 = NUMPLOT*1
      IQ3 = NUMPLOT*2
      IQ4 = NUMPLOT*3
      DO 1000 I =1,NUMPLOT
        XPLOT(I) = R*COSD(ANGLE)          /* QUADRANT 1
        YPLOT(I) = R*SIND(ANGLE)
        XPLOT(I+IQ2) =  -YPLOT(I)         /* QUADRANT 2
        YPLOT(I+IQ2) =   XPLOT(I)
        XPLOT(I+IQ3) =  -XPLOT(I)         /* QUADRANT 3
        YPLOT(I+IQ3) =  -YPLOT(I)
        XPLOT(I+IQ4) =   YPLOT(I)         /* QUADRANT 4
        YPLOT(I+IQ4) =  -XPLOT(I)

        ANGLE = ANGLE + ANGINC
1000  CONTINUE

      INCREASE = 1                        /* SET COUNTER-CLOCKWISE
      IF (IMODE.LT.0) INCREASE = -1       /* OR  CLOCKWISE

      ITEMP1 = IFIX(ANGSTART)/90
      TEMP1  = AMOD(ANGSTART,90.)
      ITEMP2 = IFIX((TEMP1*REAL(NUMPLOT))/90.)
      ISTART = ITEMP2 + (ITEMP1*NUMPLOT) +1

      ITEMP1 = IFIX(ANGEND/90.)
      TEMP1  = AMOD(ANGEND,90.)
      ITEMP2 = IFIX((TEMP1*REAL(NUMPLOT))/90.)
      IEND   = ITEMP2 + ITEMP1*(NUMPLOT) +1
```

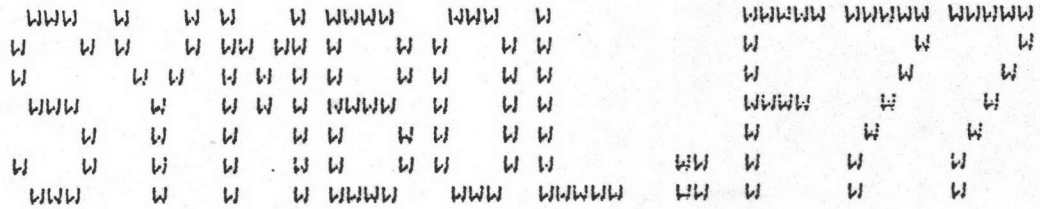
```
C
      IF   ( (ISTART.EQ.IEND).AND.
     X     ( (ANGSTART.GT.ANGEND) .OR.
     X       (ABS(AS-AE).GE.360.) ) .AND.
     X     (IMODE.GE.0) )IEND = ISTART -1
      IF   ( (ISTART.EQ.IEND) .AND.
     X     ( (ANGSTART.LT.ANGEND) .OR.
     X       (ABS(AS-AE).GE.360.) ) .AND.
     X     (IMODE.LT.0) ) IEND = ISTART+1

      IF (INCREASE.LT.0) GOTO 2000
      LOOP    = IEND-ISTART+1              /* COUNTER-CLOCKWISE
      IF (LOOP.LE.0) LOOP = LOOP + (NUMPLOT*4) + 1
      GOTO 2300
2000  CONTINUE
      LOOP    = ISTART-IEND+1              /* CLOCKWISE
      IF (LOOP.LE.0) LOOP = LOOP + (NUMPLOT*4) +1
      GOTO 2300

2300  CONTINUE
      I = ISTART
      TEMP1 = XC+XPLOT(I)
      TEMP2 = YC+YPLOT(I)
c      write(15,*)'xcircle =',temp1,temp2,i,numplot
      CALL PLOT(TEMP1,TEMP2,3)
      DO 1100 J = 1,LOOP
        I = I+INCREASE
        IF(I.LT.1)I=NUMPLOT*4
        IF(I.GT.(NUMPLOT*4)) I = 1
        CALL PLOT(XC+XPLOT(I),YC+YPLOT(I),2)
1100  CONTINUE
c      write(15,*)'xcircle ==',XC+XPLOT(LOOP),YC+YPLOT(LOOP)

9999  RETURN
      END
```

User: ECPB817203 -at PRO

<STUD>THESIS>ECPB817203>F77>CLCMTEST>SYMBOL.F77

****





****

Label: PRT015 -form

Pathname: <STUD>THESIS>ECPB817203>F77>CLCMTEST>SYMBOL.F77
File last modified: 87-08-28.15:43:48.Fri

Spooled: 87-09-18.10:37:52.Fri [Spooler rev 19.4]
Started: 87-09-18.10:55:52.Fri on: PRO by: PRO

```
C
C
C
C       PLOT A VECTORED SYMBOL
C
        SUBROUTINE SYMBOL(X,Y,HITE,ARRAY,INTEG,THETA,NCHARS)

        INTEGER*2 ARRAY(1)
        INTEGER*4 NCHARS,INTEG,IPEN
        REAL*4    X,Y,HITE,THETA,FDUMMY

        IF (NCHARS.EQ.0) NACHARS = 1
        IF (NCHARS.LT.0) GOTO 1000

          CALL XSYMBOL(X,Y,HITE,ARRAY,THETA,NCHARS)
          GOTO 9999

1000    CONTINUE
          IF (NCHARS.LT.(-2)) NCHARSS = -2
          CALL XSYMBOL(X,Y,HITE,INTEG,THETA,NCHARS)
          GOTO 9999

9999    CONTINUE
        RETURN
        END

C
C       PLOT A VECTORED SYMBOL
C
        SUBROUTINE XSYMBOL(X,Y,HITE,ARRAY,THETA,NCHARS)

        INTEGER*2 ARRAY(1)
        INTEGER*4 NCHARS,INTEG,IPEN
        REAL*4    X,Y,HITE,THETA,FDUMMY


C
C       PARAMETERS :
C                 X,Y     DEFINE THE ADDRESS OF THE LOW-LEFT CORNER OF  SYMBOL
C                 HITE    IS THE HEIGHT OF THE CHARACTERS, IN INCHES
C                 ARRAY   THE A2 ARRAY OF CHARACTERS TO BE PLOTTED
C                 THETA   THE ANGLE (IN DEGREES) FROM THE X-AXIS
C                 NCHARS # CHARACTERS TO BE PLOTTED
C


        INTEGER*2 I,IWORD,JPOS,LAST,NVECS,POS,CHAR
        REAL*4    HEIGHT,HITE2,HW,RNOCHR,RX,RY,SPACE,WIDTH,XDISP
        REAL*4    XT,YT,XSAVE,YSAVE

$INSERT LGDEVICE.F77
$INSERT CHRTABLE.F77

         IF (NCHARS.EQ.0) RETURN                    /* IGNORE BAD CALL

         TCOS=COSD(THETA)
         TSIN=SIND(THETA)
         IF(NCHARS.LT.0)GO TO 100                  /* SPECIAL SYMBOL PLOT
c         write(15,*)'symbol 2',nchars

         CALL WHERE(XT,YT,FDUMMY)
         IF ((X.EQ.999.).AND.(Y.EQ.999.)) CALL PLOT(XT,YT,3)
```

```
C
      IF ((X.EQ.999.).AND.(Y.NE.999.)) CALL PLOT(XT,Y,3)
      IF ((X.NE.999.).AND.(Y.EQ.999.)) CALL PLOT(X,YT,3)
      IF ((X.NE.999.).AND.(Y.NE.999.)) CALL PLOT(X,Y,3)

      XSAVE = X
      YSAVE = Y
      IF (X.EQ.999.) XSAVE = XT
      IF (Y.EQ.999.) YSAVE = YT

      WIDTH=.0057142B*HITE                    /* CHAR WIDTH
      HEIGHT=HITE/100.                        /* CHAR HEIGHT
      SPACE=.285714*HITE                      /* LENGTH OF SPACE BETWEEN CHARS
      RNOCHR=100.*WIDTH+SPACE                 /* NO-CHAR GAP
      XDISP=0.

C   MAIN LOOP - CYCLES ONCE EACH CHARACTER

      DO 50 I=1,NCHARS

      IWORD=(I-1)/2+1                         /* WORD # OF CURRENT CHAR
      CHAR=RS(ARRAY(IWORD),8)                 /* DEFAULT TO LEFT BYTE
      IF(RT(I,1).EQ.0)CHAR=RT(ARRAY(IWORD),8) /* UNLESS WE FIND OTHERWISE
      IF(CHAR.EQ.INTS(:240))GO TO 6               /* DON'T SCAN IF SPACE

C   SCAN THRU SYMBOL TABLE FOR CHARACTER

      POS=1
5     IF(SYM(2,POS).EQ.CHAR)GO TO 10          /* BRANCH IF FOUND
      POS=POS+SYM(3,POS)
      IF(SYM(1,POS).NE.INTS(-1))GO TO 5           /* -1 DENOTES END OF LIST

C   CHARACTER NOT FOUND - SKIP A SPACE
6     XDISP=XDISP+RNOCHR
      GO TO 50
C
C   CHARACTER FOUND
10    CONTINUE
      NVECS=SYM(3,POS)-1                      /* # VECTORS IN SYMBOL
      LAST=NVECS+POS                          /* LAST VECTOR IN TABLE TO OUTPUT
      POS=POS+1                               /* FIRST VECTOR TO OUTPUT
      CALL PLOT(XSAVE,YSAVE,3)
      CALL RELPLT(XDISP,0.,3)                 /* GO TO START-OF-CHARACTER
C
C   PLOT VECTORS
      DO 30 JPOS=POS,LAST
      RY=FLOAT(SYM(3,JPOS))*HEIGHT            /* Y-COORD FOR END OF VECTOR
      RX=FLOAT(SYM(2,JPOS))*WIDTH+SLANT*RY    /* X-COORD FOR END OF VECTOR
30    CALL RELPLT(RX,RY,INTL(SYM(1,JPOS)))        /* PLOT IT
      GO TO 6
C
50    CONTINUE
C
C   LEAVE PEN AT NEXT CHARACTER POSITION
      CALL PLOT(XSAVE,YSAVE,3)
      CALL RELPLT(XDISP,0.,3)
C
      RETURN
C
C   PLOT SPECIAL SYMBOL
100   IF(NCHARS.NE.-1.AND.NCHARS.NE.-2)GOTO 9999 /* IGNORE BAD CALL
```

```
C
      IPEN=NCHARS+4                              /* IPEN OPTION FOR 'PLOT'

      CALL WHERE(XT,YT,FDUMMY)
      IF ((X.EQ.999.).AND.(Y.EQ.999.)) CALL PLOT(XT,YT,3)
      IF ((X.EQ.999.).AND.(Y.NE.999.)) CALL PLOT(XT,Y,3)
      IF ((X.NE.999.).AND.(Y.EQ.999.)) CALL PLOT(X,YT,3)
      IF ((X.NE.999.).AND.(Y.NE.999.)) CALL PLOT(X,Y,3)
c      write(15,*)'symbol 100',nchars,ipen,x,y,xt,yt

      XSAVE = X
      YSAVE = Y
      IF (X.EQ.999.) XSAVE = XT
      IF (Y.EQ.999.) YSAVE = YT

      HITE2=HITE/2
      HW=HITE/100.

C
C     LOOKUP CHARACTER IN SYMBOL TABLE

      POS=1
105   IF(SYM(2,POS).EQ.ARRAY(2))GO TO 110
c       write(15,*)'symbol PASS',POS,SYM(2,POS)
      POS=POS+SYM(3,POS)
      IF(SYM(1,POS).NE.INTS(-1))GO TO 105
c       write(15,*)'symbol 100 RET',nchars
      GOTO 9999                                  /* NOT THERE (??)

C
C     SYMBOL FOUND...

110   CONTINUE
c       write(15,*)'symbol 100 FOUND',ARRAY(1),ARRAY(2),
      CALL RELPLT(HITE2,HITE2,3)          /* GO TO LOWER-LEFT CORNER
      NVECS=INTL(SYM(3,POS))-1                    /* # VECTORS TO PLOT
      LAST=NVECS+POS
      POS=POS+1

C
C     PLOT VECTORS
c       write(15,*)'symbol =',POS,LAST,x,y

      DO 130 JPOS=POS,LAST
        RY=FLOAT(SYM(3,JPOS))*HW
        RX=FLOAT(SYM(2,JPOS))*HW+(SLANT*RY)
c       write(15,*)'symbol ==',rx,ry,sym(1,jpos)
        CALL RELPLT(RX,RY,INTL(SYM(1,JPOS)))
130   CONTINUE

      CALL PLOT(XSAVE+HITE,YSAVE,3)                /* GO TO ORIGINAL POSITION (X,
c       write(15,*)'symbol %%%',XSAVE,YSAVE
      GOTO 9999

9999  RETURN
      END


C
C     PLOT A REAL NUMBER
C
```

```
C
      SUBROUTINE NUMBER(X,Y,HITE,RNUM,THETA,NDEC)

      INTEGER*4 NDEC,CHARS,IDUMMY
      INTEGER*2 I,NDEC2,NPLCS
      REAL*4    X,Y,HITE,RNUM,THETA
      CHARACTER*8 FMT
      CHARACTER*20 CNARRAY
C
C     PARAMETERS :
C                 X,Y    DEFINE THE COORDINATES OF THE LOWER-LEFT CORNER OF THE
C                        FIRST DIGIT TO BE PLOTTED
C                 HITE   CHARACTER HEIGHT (INCHES)
C                 RNUM   THE REAL NUMBER TO BE PLOTTED
C                 THETA  THE ANGLE (IN DEGREES) THE CHARACTER STRING MAKES WITH THE
C                        X-AXIS
C                 NDEC   THE # OF DECIMAL PLACES TO WHICH THE # IS PLOTTED
C
$INSERT LGDEVICE.F77

      NPLCS=1                                  /* DEFAULT FOR RNUM=0.

      IF(ABS(RNUM).GE.1)NPLCS=IFIX(ALOG10(ABS(RNUM)))
      NCHARS=NPLCS+INTS(NDEC)+2                        /* # CHARACTERS TO PLOT
      IF(RNUM.LE.-1.)NCHARS=NCHARS+1          /* HANDLE '-' IN NEGATIVE #
      IF(RNUM.GE.0..AND.RNUM.LT.1.)NCHARS=NCHARS-1
3     IF(NDEC.LT.0)NCHARS=NCHARS-INTS(NDEC)
      NDEC2=INTS(NDEC)
      IF (NDEC.LT.0) NDEC2=0


9     WRITE(FMT,10)NCHARS,NDEC2          /* GET FMT STMT FOR # ENCODE
10    FORMAT('(F',I2,1H.,I2,1H)')
      WRITE(CNARRAY,FMT)RNUM             /* ENCODE # INTO A2 ARRAY
      I=INTS(ICHAR(CNARRAY(1:1)))
      IF(I.EQ.INTS(:252)) GOTO 20
      IF(NDEC.LT.0)NCHARS=NCHARS+INTS(NDEC)



      CALL SYMBOL(X,Y,HITE,CNARRAY,IDUMMY,THETA,NCHARS) /* PLOT THE #
      RETURN
C
C   GUESS AT # PLACES OFF - RETRY ENCODE W/ 1 MORE PLACE
20    NCHARS=NCHARS+1
      GO TO 9
C
      END



C
C     REORGANIZE CHARACTOR TABLE
C
      SUBROUTINE REORGCHT

      INTEGER*4 POS,POS2,ITEMP1,ITEMP2

$INSERT CHRTABLE.F77
```

```
C


      POS = 1
300   IF (SYM(2,POS).EQ.INTS(9999)) GOTO 200
      POS = POS + INTL(SYM(3,POS))
      IF (SYM(1,POS).NE.INTS(-1)) GOTO 300
      GOTO 9999
200   IPOSFREE = POS
      POS2     = IPOSFREE
500   IF (SYM(2,POS2).EQ.INTS(9999)) GOTO 400
      IPOSHAVE = POS2
      DO 600 J = 1,INTL(SYM(3,POS2))
        ITEMP1 = IPOSFREE+J-1
        ITEMP2 = IPOSHAVE+J-1
        SYM(1,ITEMP1) = SYM(1,ITEMP2)
        SYM(2,ITEMP1) = SYM(2,ITEMP2)
        SYM(3,ITEMP1) = SYM(3,ITEMP2)
600   CONTINUE
      IPOSFREE = IPOSFREE + INTL(SYM(3,IPOSFREE))
      POS2     = IPOSHAVE
400   POS2     = POS2+INTL(SYM(3,POS2))
      IF (SYM(1,POS2).NE.INTS(-1)) GOTO 500
      GOTO 9999

9999  RETURN
      END
C
C
C
      SUBROUTINE USRSYM(ISYM,IOFSET,N)

      INTEGER*2  IOFSET(1)
      INTEGER*4  ISYM
      INTEGER*4  N
      INTEGER*4  POS,INCPOS,IPEN,NX,NY,PNX,PNY

*INSERT CHRTABLE.F77

c      write(15,*)'usrsym',isym,n

      IF (N.EQ.-1001) GOTO 1000
      IF (N.EQ.-1002) GOTO 2000
      IF ((N.LT.1).OR.(N.GT.1000)) GOTO 9999
      POS = 1

200   IF (SYM(2,POS).NE.INTS(ISYM)) GOTO 100
      SYM(2,POS) = 9999                    /* DELETE OLD SYMBOL
100   POS = POS + INTL(SYM(3,POS))
      IF (SYM(1,POS).NE.INTS(-1)) GOTO 200

C
C     ASSIGN USER SYMBOL INTO THE CHARACTER TABLE

c      write(15,*)'usrsym write'
      SYM(1,POS) = INTS(0)
      SYM(2,POS) = INTS(ISYM)
      INCPOS = 0
      IPEN = 3
      PNX  = 0
      PNY  = 0
      DO 300 J = 1,N
        NX  = INTL(IOFSET(J)/INTS(100))
```

```
C

        NY  = INTL(MOD(INTL(IOFSET(J)),100))
C        write(15,*)'usrsym>',nx,ny,IOFSET(J)
        IF ((NX.LT.0).OR.(NX.GT.15)) NX = 0
        IF ((NY.LT.0).OR.(NY.GT.15)) NY = 0
        IF ((NX.EQ.15).AND.(NY.EQ.15)) GOTO 300
        IF ((NX.EQ.15).AND.(NY.EQ.0))  IPEN = 3
        IF ((NX.EQ.15).AND.(NY.EQ.0))  GOTO 300
        INCPOS = INCPOS +1
        SYM(1,POS+INCPOS) = INTS(IPEN)
        SYM(2,POS+INCPOS) = INTS(REAL(NX-PNX)*100./15.)
        SYM(3,POS+INCPOS) = INTS(REAL(NY-PNY)*100./15.)
C        write(15,*)'usrsym',sym(1,pos+incpos),sym(2,pos+incpos),
C     X              sym(3,pos+incpos),ipen
        IPEN = 2
        PNX = NX
        PNY = NY
300   CONTINUE
      SYM(3,POS) = INTS(INCPOS+1)
      GOTO 9999

1000  CONTINUE
      POS = 1
1200  IF ((SYM(2,POS).GE.INTS(288)) .OR.
     X    (SYM(2,POS).LE.INTS(383))) SYM(2,POS)=INTS(9999)
      POS = POS + INTL(SYM(3,POS))
      IF (SYM(1,POS).NE.INTS(-1)) GOTO 1200
      GOTO 3000
2000  CONTINUE
      POS = 1
2200  IF ((SYM(2,POS).GE.INTS(384)) .OR.
     X    (SYM(2,POS).LE.INTS(479))) SYM(2,POS) = INTS(9999)
      POS = POS + INTL(SYM(3,POS))
      IF (SYM(1,POS).NE.INTS(-1)) GOTO 2200
      GOTO 3000

3000  CONTINUE
      CALL REORGCHT
      GOTO 9999

9999  CONTINUE
      RETURN
      END
```

ประวัติผู้เขียน

นายวิทวัส จารุจันทร เกิดวันที่ 17 สิงหาคม 2506 ที่กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาตรบัณฑิต จากจุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2527 และเข้าศึกษาต่อในระดับปริญญามหาบัณฑิตของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2528