

ลักษณะของโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม

3.1 ลักษณะทั่วไปของโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม

ในบทนี้จะอธิบายวิธีเรียกใช้โปรแกรมย่อยต่างๆ สำหรับผู้ใช้ที่เขียนโปรแกรมโดยใช้ภาษาฟอร์แทรน เพราะโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรมถูกพัฒนาโดยใช้ภาษาฟอร์แทรน

ลักษณะของโปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม จะถูกแบ่งออกเป็น 3 ลักษณะ คือ

(คำว่า เวกเตอร์ หมายถึง คำสั่งในการพลอตซึ่งอาจเป็นการลากเส้น หรือเพียงแต่ย้ายตำแหน่งการพลอต อย่างไม่อย่างหนึ่ง)

(คำว่า พารามิเตอร์ หมายถึง ข้อมูลที่ส่งให้โปรแกรมย่อยเมื่อมีการเรียกใช้)

ลักษณะแรก ประกอบด้วยโปรแกรมย่อยที่ทำหน้าที่จัดการอุปกรณ์แสดงผลลัพท์ทางกราฟฟิก และจัดเตรียมแฟ้มข้อมูลที่เก็บเวกเตอร์การพลอตและแฟ้มข้อมูลที่เก็บข้อผิดพลาดการพลอต(พลอตเกินขอบเขตที่กำหนด) และกำหนดค่าเริ่มต้นสำหรับตัวแปรร่วมที่ใช้ควบคุมโปรแกรมย่อยอื่นๆ

ลักษณะที่สอง ประกอบด้วยโปรแกรมย่อยที่ทำหน้าที่พื้นฐานในการพลอตเช่น

- โปรแกรมย่อย PLOTS สำหรับเริ่มต้นการเรียกใช้โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม
- โปรแกรมย่อย PLOT สำหรับพลอตเส้นหรือย้ายตำแหน่งพลอต
- โปรแกรมย่อย WHERE สำหรับตรวจสอบตำแหน่งสุดท้ายที่ทำการพลอตและอัตราย่อขยายขนาดภาพ
- ฯลฯ

ซึ่งลักษณะที่สองนี้ได้ถูกจัดทำขึ้นให้มีวิธีเรียกใช้โปรแกรมย่อยเหมือนกับโปรแกรมย่อยที่ใช้แสดงผลลัพท์ทางพลอตเตอร์ ' ' ทำให้โปรแกรมสำหรับงานวิศวกรรมที่มีอยู่แล้ว

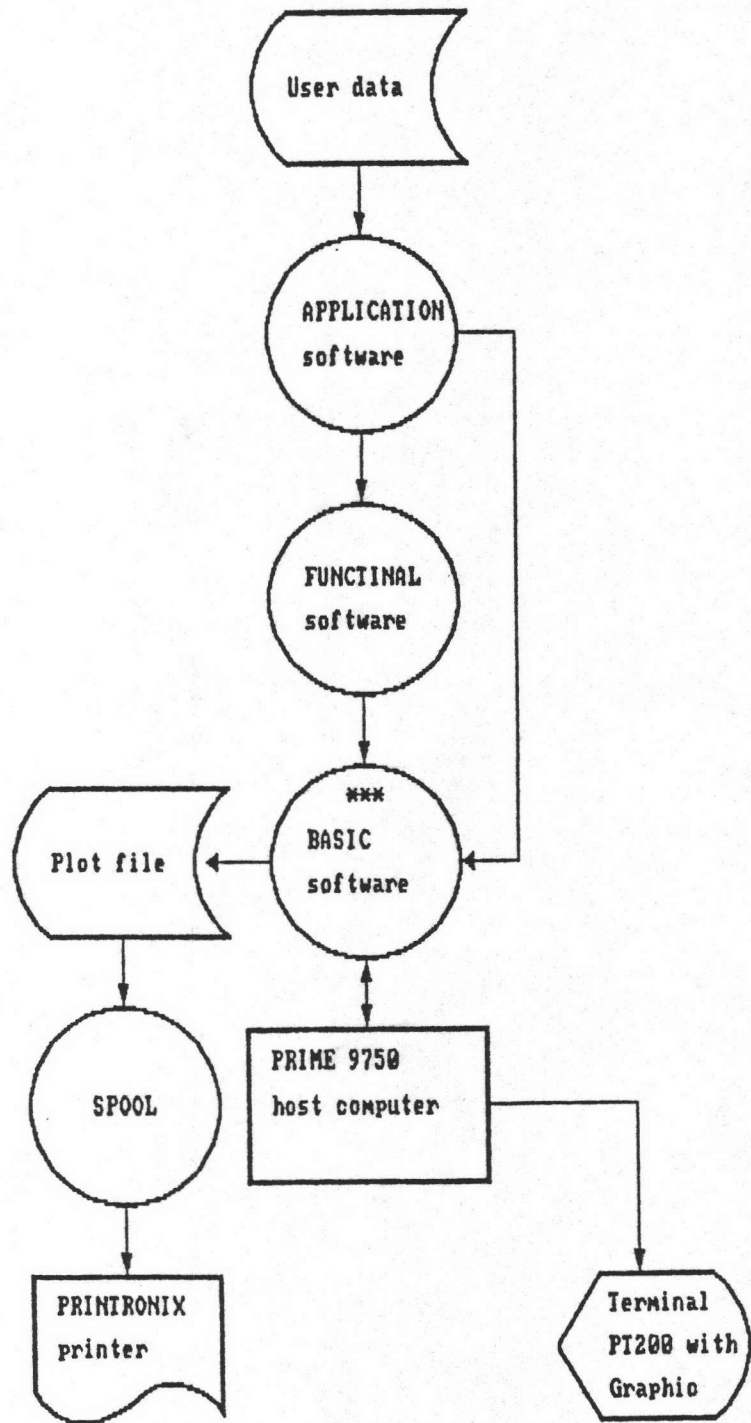
ซึ่งแสดงผลลัพธ์ได้ทางพลอตเตอร์ทางเดียว สามารถแสดงผลลัพธ์ทางเครื่องพิมพ์ และจอภาพด้วย เหมาะสำหรับทดสอบดูผลลัพธ์ที่แสดงทางจอภาพหรือเครื่องพิมพ์ ให้ถูกต้องก่อนจึงแสดงผลลัพธ์ทางพลอตเตอร์ ทำให้ประหยัดกระดาษและหมึกที่ใช้ ในการพลอตทางพลอตเตอร์ซึ่งมีต้นทุนสูงกว่า แสดงผลลัพธ์ทางจอภาพและเครื่องพิมพ์ และใช้เวลามากกว่าด้วย

ลักษณะที่สามจะประกอบด้วยโปรแกรมย่อยที่ทำหน้าที่พิเศษเฉพาะอย่าง สำหรับผู้ใช้ เช่น

- โปรแกรมย่อย PIE สำหรับพลอตกราฟวงกลม
- โปรแกรมย่อย BAR สำหรับพลอตกราฟแท่ง
- โปรแกรมย่อย AROHD สำหรับพลอตรูปลูกศร
- ฯลฯ

โปรแกรมย่อยทางกราฟฟิกสำหรับงานวิศวกรรม สามารถช่วยให้ผู้ใช้ที่มีความต้องการแสดงผลลัพธ์ทางกราฟฟิก สามารถพัฒนาโปรแกรมขึ้นมาโดยอาศัยการเรียกใช้โปรแกรมย่อยทางกราฟฟิกได้อย่างมีประสิทธิภาพ และสามารถพลอตกราฟ ลักษณะต่างๆ เช่น กราฟเส้น กราฟแท่ง กราฟวงกลม ฯลฯ อีกทั้งยังลดความยุ่งยาก จากการจัดการกับอุปกรณ์แสดงผลลัพธ์ทางกราฟฟิก โดยจะมีโปรแกรมย่อยที่จัดการ เรื่องนี้แทนให้ รูปที่ 3-1 จะแสดงลักษณะทั่วไปของโปรแกรมย่อยทางกราฟฟิก สำหรับงานวิศวกรรม ในส่วนต่อไปจะอธิบายหน้าที่และรูปแบบการเรียกใช้ของแต่ละ โปรแกรมย่อย

(ตัวอย่างการเรียกใช้โปรแกรมย่อยทางกราฟฟิก สามารถดูได้จากภาคผนวก ก.)



รูปที่ ๑-๑ แสดงลักษณะทั่วไปของโปรแกรมย่อยทางกราฟิกสำหรับงานวิศวกรรม

3.2 โปรแกรมย่อยที่มีหน้าที่เกี่ยวกับการจัดการอุปกรณ์ที่ใช้แสดงผลลัพท์ หรือ ควบคุมการทำงานของโปรแกรมย่อยอื่นๆ

3.2.1 โปรแกรมย่อย INITPARA (INITIALize PARAMETERS)

ใช้สำหรับตอบโต้กับผู้ใช้ตอนเริ่มต้นการเรียกใช้โปรแกรมย่อยทางกราฟฟิก สำหรับงานวิศวกรรมโดยโปรแกรมหลักของผู้ใช้ ทั้งนี้เพื่อให้ผู้ใช้สามารถกำหนด การทำงานของโปรแกรมย่อยทางกราฟฟิกให้เป็นไปตามความต้องการของผู้ใช้ได้ ในลักษณะต่างๆ ดังนี้

(คำว่า เวกเตอร์ หมายถึง คำสั่งในการพลอตซึ่งอาจเป็นการลากเส้น หรือเพียง แต่ย้ายตำแหน่งการพลอต ใดๆอย่างหนึ่ง)

1) สามารถกำหนดอุปกรณ์ที่จะใช้แสดงผลลัพท์ทางกราฟฟิกโดย โปรแกรมย่อยนี้จะถามว่า ต้องการแสดงผลลัพท์ทางจอภาพหรือแฟ้มข้อมูลซึ่งจะเก็บ ข้อมูลในรูปของเวกเตอร์ของการพลอต ประกอบด้วยตำแหน่งตามแนวราบจริง ของจอภาพ (ค่า X ตามแกน-X) และตำแหน่งตามแนวตั้งจริงของจอภาพ (ค่า Y ตามแกน-Y) โดยตำแหน่งตามแนวตั้งนี้จะมีค่าอ้างอิงตั้งแต่ 0 ถึง 1023 และ ตำแหน่งตามแนวตั้งจะมีค่าอ้างอิงตั้งแต่ 0 ถึง 779 รูปแบบโครงสร้างของข้อมูลที่ เก็บในแฟ้มข้อมูลคือ ค่า VX และค่า VY โดย VX จะมีค่าเท่ากับ X และค่าสัมบูรณ์ ของ VY จะมีค่าเท่ากับ $Y + 1$ และ ถ้า VY มีค่าเป็นลบ หมายความว่า มีการ เปลี่ยนตำแหน่งพลอตโดยไม่มีการลากเส้น ส่วนถ้า VY มีค่าเป็นบวก หมายความว่า มีการเปลี่ยนตำแหน่งพลอตโดยมีการลากเส้น สาเหตุที่ต้องทำการเก็บค่า $V + 1$ เพราะในกรณีที่ Y มีค่าเท่ากับ 0 จะทำให้ ค่าเครื่องหมายบวกลบเก็บไม่ได้เนื่อง จาก 0 ไม่มีค่าลบ คำถามที่ถามผู้ใช้จะปรากฏดังนี้

SELECT PLOT MODE T = TERMINAL, F = FILE :

(ผู้ใช้ต้องเลือกกดปุ่ม T หรือ F ตามความต้องการของผู้ใช้)

2) สามารถกำหนดว่า หนึ่งหน่วยความยาวที่ผู้ใช้ใช้ในการพลอตภาพเช่น นิ้ว เซนติเมตร กิโลเมตร ฯลฯ จะแทนด้วยจำนวนจุดกี่จุด ทั้งนี้หมายถึงความยาวแนวราบและแนวดิ่ง ไม่ใช่แนวเฉียง ซึ่งจำนวนจุดต่อหนึ่งหน่วยความยาวนี้จะแทนด้วยตัวแปรร่วม RPIX และ RPIY (RPI = Raster Per Inch) ซึ่ง RPIX มีค่าเท่ากับ RPIY เสมอจำนวนจุดต่อหนึ่งหน่วยความยาวนี้มีให้ผู้ใช้เลือกได้ 9 จำนวน ตั้งแต่ 5 จุดต่อหนึ่งหน่วยความยาวจนถึง 800 จุดต่อหนึ่งหน่วยความยาว โดยคำถามผู้ใช้จะปรากฏดังนี้

SCREEN HAS 700 X 1000 DOTS

- | | | |
|----|-----|-----|
| 1. | 5 | RPI |
| 2. | 10 | RPI |
| 3. | 15 | RPI |
| 4. | 25 | RPI |
| 5. | 50 | RPI |
| 6. | 100 | RPI |
| 7. | 200 | RPI |
| 8. | 400 | RPI |
| 9. | 800 | RPI |

SELECT :

3) สามารถกำหนดได้ว่าต้องการให้นำคำสั่งพลอตไปพลอตทันที (เลือก I) หรือเก็บในบัฟเฟอร์ก่อนเมื่อบัฟเฟอร์เต็มจึงค่อยทำการพลอต (เลือก B) โดยคำถามที่ถามผู้ใช้จะปรากฏดังนี้ (คำว่า บัฟเฟอร์ หมายถึง ที่ว่างซึ่งใช้เก็บเวกเตอร์การพลอตชั่วคราว)

SELECT TERMINAL PLOT MODE B = BUFFER, I = IMMEDIATELY :

4) สามารถกำหนดชื่อของแฟ้มข้อมูล สำหรับเก็บเวกเตอร์การพลอต ในกรณีที่ต้องการพลอตลงแฟ้มข้อมูล ถ้าผู้ใช้ไม่กำหนดชื่อแฟ้มข้อมูลมาโปรแกรมย่อยนี้ จะกำหนดให้ชื่อว่า VECTOR โดยคำถามที่จะถามผู้ใช้จะปรากฏดังนี้

ENTER VECTOR FILENAME :

5) สามารถกำหนดชื่อของแฟ้มข้อมูล สำหรับเก็บข้อผิดพลาดการพลอต ออกนอกขอบเขตของจอภาพ ซึ่งถ้าผู้ใช้ไม่กำหนดชื่อแฟ้มข้อมูลมาโปรแกรมย่อยนี้ จะกำหนดให้ชื่อว่า ERROR โดยคำถามที่ถามผู้ใช้จะปรากฏดังนี้

ENTER ERROR OUTPUT FILENAME :

6) สามารถกำหนดได้ว่าแนวราบของรูปภาพจะขนานกับแนวราบของจอภาพ (HORIZONTAL) หรือ จะตั้งฉากกับแนวราบของจอภาพ (VERTICAL) โดยกำหนด คำถามที่ถามผู้ใช้จะปรากฏดังนี้

SELECT AXIS MODE H = HORIZONTAL, = V = VERTICAL :

รูปแบบการเรียกใช้โปรแกรมย่อย INITPARA

CALL INITPARA

(ไม่ต้องมีพารามิเตอร์ที่ต้องส่งค่าเข้ามา)

3.2.2 โปรแกรมย่อย INITSYST (INITialize SYSTem)

ใช้สำหรับกำหนดค่าเริ่มต้นให้กับตัวแปรร่วมที่ใช้ควบคุมโปรแกรมย่อยอื่น ซึ่งไม่จำเป็นต้องให้ผู้ใช้เป็นผู้กำหนด เช่น

- กำหนดตำแหน่งของจุดกำเนิดจะเริ่มไว้ที่ตำแหน่ง (0,0)
- กำหนดตำแหน่งสุดท้ายที่มีการพลอตจะไว้ที่ตำแหน่ง (0,0)
- กำหนดมุมเอียงภาพให้เริ่มต้นเท่ากับ 0 องศา (ไม่มีการหมุน)
- กำหนดอัตราย่อขยายขนาดภาพให้เริ่มต้นเท่ากับ 1 (ขนาดปกติ)
- กำหนดมุมเอียงตัวอักษรให้เริ่มต้นเท่ากับ 0 องศา (ตัวอักษรไม่เอียง)
- ฯลฯ

และสำหรับจัดการอุปกรณ์แสดงผลทางกราฟิกเช่น ในกรณีที่ต้องการแสดงผลทางกราฟิกบนจอภาพก็ต้องมี การเปลี่ยนโหมดของจอภาพและการล้างจอภาพ และในกรณีที่ต้องการแสดงผลทางกราฟิกบนเครื่องพิมพ์ ซึ่งต้องมีการเก็บแวกเตอร์การพลอตภาพไว้ในแฟ้มข้อมูลก่อน ซึ่งแฟ้มข้อมูลนี้ต้องมีการกำหนดรูปแบบโครงสร้างของข้อมูลที่เก็บและทำการเปิดแฟ้มข้อมูลด้วย

หลังจากโปรแกรมย่อยนี้ถูกเรียกใช้ (ใช้ร่วมกับโปรแกรมย่อย INITPARA ตอนเริ่มเรียกใช้โปรแกรมย่อยทางกราฟิกสำหรับงานวิศวกรรม) โปรแกรมย่อยทางกราฟิกอื่นๆจึงสามารถทำงานได้ ซึ่งโปรแกรมย่อยทั้งสองนี้จะถูกเรียกใช้โดยโปรแกรมย่อย PLOTS อีกต่อหนึ่ง

รูปแบบการเรียกใช้โปรแกรมย่อย INITSYST

CALL INITSYST

(ไม่ต้องมีพารามิเตอร์ที่ต้องส่งค่าเข้ามา)

3.2.3 โปรแกรมย่อย ENDPLT (END PLOT)

ใช้สำหรับกรณีที่ใช้เลิกการเรียกใช้โปรแกรมย่อยทางกราฟิกสำหรับงานวิศวกรรม โปรแกรมย่อยนี้จะทำการเก็บข้อมูลบนหัวของแฟ้มข้อมูลที่เก็บเวกเตอร์การพลอต ทำการปิดแฟ้มข้อมูล (ในกรณีที่มีความต้องการแสดงผลลัพท์ทางกราฟิกบนแฟ้มข้อมูล) หรือจะทำการเปลี่ยนโหมดของจอภาพจากโหมดกราฟิกกลับคืนเป็นโหมดตัวอักษรสำหรับทำงานตามปกติ (ในกรณีที่มีความต้องการแสดงผลลัพท์ทางกราฟิกบนจอภาพ) โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมย่อย PLOT โดยกำหนดค่า IPEN ให้เท่ากับ 999

รูปแบบการเรียกใช้โปรแกรมย่อย ENDPLT

CALL ENDPLT

(ไม่ต้องมีพารามิเตอร์ที่ต้องส่งค่าเข้ามา)

3.2.4 โปรแกรมย่อย CLRBUF (CLear BUfFer)

ใช้สำหรับจัดการเกี่ยวกับคำสั่งพลอตที่มาจากโปรแกรมย่อย PLOT ซึ่งจะอยู่ในรูปเวกเตอร์การพลอต ซึ่งคำสั่งพลอตนี้จะเก็บอยู่ในบัฟเฟอร์ก่อน โปรแกรมย่อยนี้จะนำคำสั่งพลอตในบัฟเฟอร์ไปแสดงผลลัพท์ เช่น ในกรณีที่ต้องการแสดงผลลัพท์ทางกราฟิกบนแฟ้มข้อมูล โปรแกรมย่อยนี้จะทำการอ่านข้อมูลจากบัฟเฟอร์มาบันทึกลงในแฟ้มข้อมูลที่เก็บเวกเตอร์การพลอต ซึ่งแฟ้มข้อมูลนี้จะถูกนำพลอตยังเครื่องพิมพ์ต่อไป โปรแกรมย่อยนี้จะถูกเรียกใช้ในกรณีที่บัฟเฟอร์เต็มหรือเลิกการเรียกใช้โปรแกรมย่อยทางกราฟิกสำหรับงานวิศวกรรม

รูปแบบการเรียกใช้โปรแกรมย่อย CLRBUF

CALL CLRBUF

(ไม่ต้องมีพารามิเตอร์ที่ต้องส่งค่าเข้ามา)

3.2.5 โปรแกรมย่อย GENFPLOT

ใช้สำหรับนำเวกเตอร์ของการพลอตในแฟ้มข้อมูลที่เก็บเวกเตอร์การพลอตมาเปลี่ยนรูปแบบโครงสร้างของข้อมูลมาอยู่ในลักษณะหนึ่งจุดต่อหนึ่งบิต (bit map) แล้วทำการใส่คำสั่งสำหรับสั่งให้เครื่องพิมพ์ทำการพิมพ์ในโหมดกราฟฟิกด้วย ลงในแฟ้มข้อมูลที่เปรียบเสมือนกระดาษ (plot file) และทำการส่งแฟ้มข้อมูลนี้ไปพิมพ์ยังเครื่องพิมพ์ต่อไป

รูปแบบการเรียกใช้โปรแกรมย่อย GENFPLOT

```
CALL GENFPLOT
```

(ไม่ต้องมีพารามิเตอร์ที่ต้องส่งค่าเข้ามา)

3.2.6 โปรแกรมย่อย BDRWLINE (BEST DRAW LINE)

ใช้สำหรับลากเส้นบนกระดาษจำลองโดยกำหนดตำแหน่งเริ่มต้นของเส้นตรงที่จะทำการพลอต (ตำแหน่งอ้างอิงนี้คือตำแหน่งบนจอภาพจริง ไม่ใช่ตำแหน่งอ้างอิงที่ผู้ใช้ใช้อยู่) และตำแหน่งสิ้นสุดของเส้นตรงที่จะทำการพลอตภาพ โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมย่อย GENFPLOT

รูปแบบการเรียกใช้โปรแกรมย่อย BDRWLINE

```
INTEGER*4 IX1, IY1, IX2, IY2
```

```
CALL BDRWLINE (IX1, IY1, IX2, IY2)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร IX1, IY1 ตำแหน่งเริ่มต้นของเส้นตรงที่จะทำการพลอต

ตัวแปร IX2, IY2 ตำแหน่งสิ้นสุดของเส้นตรงที่จะทำการพลอต

3.3 โปรแกรมย่อยทางกราฟฟิกที่มีหน้าที่การทำงานพื้นฐาน

3.3.1 โปรแกรมย่อย PLOTS

ใช้ในตอนเริ่มต้นเรียกใช้ชุดโปรแกรมย่อยกราฟฟิก แล้วจึงจะทำการเรียกใช้โปรแกรมย่อยอื่นภายในชุดโปรแกรมย่อยกราฟฟิกได้ โปรแกรมย่อยนี้จะเรียกใช้โปรแกรมย่อย INITPARA และ INITSYST อีกต่อหนึ่ง

รูปแบบการเรียกใช้โปรแกรมย่อย PLOTS

```
INTEGER*4 IDUMMY1, IDUMMY2, IDUMMY3
CALL PLOTS (IDUMMY1, IDUMMY2, IDUMMY3)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร IDUMMY1, IDUMMY2, IDUMMY3 เป็นตัวแปรที่ตอนนี้สงวนไว้
ใช้ต่อไปในอนาคต

3.3.2 โปรแกรมย่อย NEWPLT

ใช้ในกรณีที่ต้องการพลอตรูปหลายๆ รูปแยกจากกันแต่ต้องการทำภายในโปรแกรมเดียวกัน เมื่อพลอตรูปหนึ่งเสร็จก็ออกผลลัพธ์แล้วก็จะทำการพลอตภาพต่อไปเหมือนเรียกใช้โปรแกรมย่อย PLOTS อีกครั้งหนึ่ง

รูปแบบการเรียกชุดโปรแกรมย่อย NEWPLT

```
CALL NEWPLT
(ไม่ต้องมีพารามิเตอร์ที่ต้องส่งค่าเข้ามา)
```

3.3.3 โปรแกรมย่อย PLOT

ใช้สำหรับย้ายตำแหน่งของจุดที่จะทำการพลอตมี 2 แบบ คือย้ายตำแหน่งไปอย่างเดียวไม่มีการลากเส้นจากจุดเดิมที่เคยอยู่มาด้วย อีกแบบย้ายตำแหน่งโดยมีการลากเส้นไปด้วยจากจุดเดิมที่เคยอยู่ ถ้าเทียบกับเครื่องพลอตเตอร์แล้วก็คือการย้ายตำแหน่งของปลายปากกาโดยทำการยกปากกากระหว่างมีการย้ายตำแหน่ง อีกแบบคือการย้ายปากกาโดยปากกายังคงลากอยู่บนกระดาษที่ทำการพลอต

รูปแบบการเรียกใช้โปรแกรมย่อย PLOT

```
INTEGER*4 X,Y
INTEGER*4 IPEN
CALL PLOT (X,Y,IPEN)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X,Y ใช้บอกตำแหน่งของจุดที่ต้องการย้ายไปทำการพลอต โดยตำแหน่งนี้ต้องอ้างอิงจากจุดกำเนิดที่กำหนดไว้

ตัวแปร IPEN ถ้ามีค่าเป็นลบหมายความว่าให้ตำแหน่งที่กำหนดมาจะถูกกำหนดเป็นจุดกำเนิดใหม่ใช้อ้างอิงตำแหน่งต่อไป ส่วนถ้าค่าเป็นบวกจะไม่มีมีการกำหนดจุดกำเนิดใหม่ ให้ใช้จุดกำเนิดที่มีอยู่เดิมต่อไป และค่าลัมบ์ของ IPEN นี้จะมีเพียง 3 ค่า คือ

IPEN มีค่าเท่ากับ 2 หมายความว่า การย้ายตำแหน่งนี้ให้มีการลากเส้นจากจุดล่าสุดมายังจุดที่กำหนดมาด้วย

IPEN มีค่าเท่ากับ 3 หมายความว่า มีแต่การย้ายตำแหน่งอย่างเดียว

IPEN มีค่าเท่ากับ 999 หมายความว่า มีการพลอตรูปได้สิ้นสุดลงแล้ว จะเลิกการเรียกใช้โปรแกรมย่อยทางกราฟฟิก

3.3.4 โปรแกรมย่อย FACTOR

ใช้กำหนดอัตราย่อขยายขนาดภาพที่จะทำการพลอต โดยเปลี่ยนอัตราส่วนของหน่วยความยาวที่ใช้ เช่น ถ้าส่งค่า 0.5 ภาพที่ได้จะมีขนาดเล็กลงเป็นครึ่งหนึ่ง หรือถ้าส่งค่า 2. ภาพที่ได้จะมีขนาดเพิ่มขึ้นเป็น 2 เท่า

ข้อแนะนำ ในระหว่างที่ทำการทดสอบโปรแกรมที่ใช้งานอยู่ซึ่งต้องมีการเรียกโปรแกรมมาทำงานหลายครั้ง ถ้ามีการลดขนาดของรูปลงจะทำให้การทำงานของโปรแกรมเร็วขึ้นเนื่องจากจุดที่จะทำการพลอตน้อยลง เมื่อทดสอบโปรแกรมเรียบร้อยแล้วจึงกำหนดค่า FACT เพื่อให้ได้รูปตามปกติต่อไป

รูปแบบการเรียกใช้โปรแกรมย่อย FACTOR

```
REAL*4      FACT
CALL FACTOR(FACT)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร FACT ใช้กำหนดอัตราย่อขยายขนาดภาพเช่น

ถ้ากำหนดค่า FACT = 3.0 เวลาที่มีการอ้างอิงถึงความยาวจะถูกนำมาคูณกับค่า FACT นี้ซึ่งจะได้ค่าเพิ่มขึ้นเป็น 3 เท่า

ถ้ากำหนดค่า FACT = 1.0 ขนาดของความยาวจะเป็นปกติตามที่มีการใช้

3.3.5 โปรแกรมย่อย WHERE

ใช้สำหรับหาตำแหน่งล่าสุดที่มีการพลอต และค่า FACT ที่ใช้กำหนด อัตราย่อขยายขนาดภาพ ใช้ในกรณีที่มีผู้เขียนโปรแกรมเดียวกันร่วมกันหลายคน

รูปแบบการเรียกใช้โปรแกรมย่อย WHERE

```
REAL*4      X,Y
REAL*4      FACT
CALL WHERE (X,Y, FACT)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X,Y ใช้บอกตำแหน่งล่าสุดที่มีการพลอต ซึ่งเกิดจากการเรียกใช้โปรแกรมย่อย PLOT ก่อนหน้านี้

ตัวแปร FACT บอกขนาดอัตราย่อขยายขนาดภาพ (ค่านี้จะถูกเริ่มต้นด้วย 1.0 ในตอนเริ่มต้นการเรียกใช้โปรแกรมย่อยทางกราฟฟิก)

3.3.6 โปรแกรมย่อย ROTATE

ใช้กำหนดมุมเอียงภาพ

ข้อแนะนำ สำหรับกรณีที่มีความต้องการหมุนภาพให้เอียงตามที่กำหนด ส่วนของโปรแกรมที่มีการเรียกใช้โปรแกรมย่อย PLOT ต้องแทนด้วยการเรียกใช้โปรแกรมย่อย RELPLT(โปรดดูวิธีเรียกใช้โปรแกรมย่อย RELPLT) พร้อมกับการเรียกใช้โปรแกรมย่อย WHERE ดังตัวอย่างต่อไปนี้

```

REAL*4    X,Y
INTEGER*4 IPEN
CALL PLOT(X,Y,IPEN)
แทนด้วย
REAL*4    X,Y, LASTX, LASTY, DUMMY
INTEGER*4 IPEN
CALL WHERE(LASTX, LASTY, DUMMY)
CALL RELPLT(X-LASTX, Y-LASTY, IPEN)

```

รูปแบบการเรียกใช้โปรแกรมย่อยนี้

```

REAL * 4 ANGLE
CALL ROTATE (ANGLE)

```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร ANGLE คือมุมที่อ้างอิงกับแนวราบของจอภาพใช้กำหนดมุมเอียงของเวกเตอร์ที่ถูกพลอตผ่านโปรแกรมย่อย RELPLT

3.3.7 โปรแกรมย่อย MIRROR

ใช้สำหรับการสะท้อนภาพที่พลอต (สะท้อนตำแหน่งที่ใช้พลอต ให้กลับซ้ายเป็นขวา หรือบนเป็นล่าง หรือทั้งสองอย่าง)

รูปแบบการเรียกใช้โปรแกรมย่อย MIRROR

```
INTEGER*4      ITYPE
CALL MIRROR ( ITYPE )
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร ITYPE

- | | |
|-------------------|---------------------------------------|
| ถ้ามีค่าเท่ากับ 0 | รูปจะเป็นปกติ |
| ถ้ามีค่าเท่ากับ 1 | ตำแหน่งแนวราบสะท้อน (กลับซ้ายเป็นขวา) |
| ถ้ามีค่าเท่ากับ 2 | ตำแหน่งแนวตั้งสะท้อน (กลับบนเป็นล่าง) |
| ถ้ามีค่าเท่ากับ 3 | ตำแหน่งสะท้อนทั้งสองแกน |

3.3.8 โปรแกรมย่อย SYMBOL

ใช้พลอตตัวอักษรต่างๆของรหัส ASCII ซึ่งตัวอักษร ASCII จะประกอบด้วย ตัวอักษรใหญ่ A ถึง Z ตัวอักษรเล็ก a ถึง z และตัวอักษรอื่นๆ เช่น * + - ฯลฯ อีกทั้งยังประกอบไปด้วยตัวอักษรพิเศษใช้สำหรับพลอตกราฟ และตัวอักษรที่ผู้ใช้กำหนดสร้างขึ้นมาเองด้วย

รูปแบบการเรียกใช้โปรแกรมย่อย SYMBOL

```
REAL*4      X, Y, HITE, ANGLE
INTEGER*4   ICHRCODE, NCHAR
CHARACTER*n CHRSTRING
CALL SYMBOL (X, Y, HITE, CHRSTRING, ICHRCODE, ANGLE
            *, NCHAR)
```

(n คือจำนวนตัวอักษรของแถวตัวอักษรที่ผู้ใช้ต้องการใช้ซึ่งผู้ใช้ต้องการซึ่งผู้ใช้ต้องกำหนดเอง)

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X, Y กำหนดตำแหน่งเริ่มต้นที่จะทำการพลอตแถวตัวอักษร คือ มุมล่างซ้ายสุดของตัวอักษรตัวแรกของแถวตัวอักษร

ถ้า X มีค่าเท่ากับ 999. จะใช้ค่า X ล่าสุดที่มีการพลอต

ถ้า Y มีค่าเท่ากับ 999. จะใช้ค่า Y ล่าสุดที่มีการพลอต

ตัวแปร HITE ใช้กำหนดความสูงของตัวอักษร

ตัวแปร CHRSTRING ใช้ในกรณี NCHAR มีค่ามากกว่าศูนย์ สำหรับเก็บแถวตัวอักษรที่จะทำการพลอตใช้

ตัวแปร ICHRCODE ใช้เก็บรหัสตัวอักษรพิเศษหรือรหัสของตัวอักษรที่ผู้ใช้สร้างขึ้นเอง ใช้ในกรณีที่ NCHAR มีค่าเท่ากับ -1 หรือ -2 เท่านั้น

ตัวแปร ANGLE ใช้กำหนดมุมเอียงของแถวตัวอักษรที่จะทำการพลอตทั้งแถว (หน่วยที่ใช้คือ องศา) โดยอ้างอิงแกนแนวราบของจอภาพเป็นมุมศูนย์องศา

ถ้า ANGLE = 0. องศา หมายความว่า จะมีการพลอต แถวตัวอักษรตามปกติ เช่นการเขียนตามปกติ

ถ้าค่า ANGLE มากกว่าหรือเท่ากับ 360. จะถูกนำมาหารเป็นเศษที่มีค่าน้อยกว่า 360. และมากกว่าหรือเท่ากับ 0.

ถ้าค่า ANGLE น้อยกว่า 0. จะถูกนำมาหารเป็นเศษที่มีค่าน้อยกว่า 360. และมากกว่าหรือเท่ากับ 0.

ตัวแปร NCHAR ใช้บอกจำนวนตัวอักษรที่ทำการพลอต

ถ้าค่า NCHAR มีค่ามากกว่าศูนย์ NCHAR คือจำนวนตัวอักษร

ถ้าค่า NCHAR มีค่าเท่ากับศูนย์ตัวอักษรตัวแรกของแถวตัวอักษรจะถูกนำมาพลอต

ถ้าค่า NCHAR มีค่าเป็นลบ จะทำการพลอตตัวอักษรพิเศษที่ใช้พลอตกราฟหรือ ตัวอักษรที่ผู้ใช้สร้างขึ้น

ถ้า NCHAR มีค่าเท่ากับ -1 จะไม่มีการลากเส้นจากตำแหน่งล่าสุดมายังตำแหน่ง X,Y ที่กำหนดมาข้างต้น

ถ้า NCHAR มีค่าเท่ากับ -2 หมายความว่า จะมีการลากเส้นจากตำแหน่งล่าสุดที่มีการพลอตมายังตำแหน่ง X,Y ที่กำหนดมาข้างต้น



3.3.9 โปรแกรมย่อย CSLANT

ใช้สำหรับกำหนดมุมเอียงของตัวอักษรที่จะทำการพลอตผ่านโปรแกรมย่อย

SYMBOL

รูปแบบการเรียกใช้โปรแกรมย่อย CSLANT

```
REAL*4 SL
```

```
CALL CSLANT(SL)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร SL กำหนดมุมเอียงของตัวอักษรหน่วยเป็นองศา

3.3.10 โปรแกรมย่อย USRSYM

ใช้สำหรับให้ผู้ใช้สร้างตัวอักษรขึ้นมาเอง ตามความต้องการของผู้ใช้ โดยตัวอักษรที่สร้างมี ลักษณะเป็นเวกเตอร์ (VECTOR) โดยจำนวนเวกเตอร์ใช้ ได้มากที่สุดคือ 1,000 เวกเตอร์ (รวมทั้งหมดของทุกตัวอักษรถ้าใช้ไปแล้ว 1,000 เวกเตอร์แล้วก็จะสร้างตัวอักษรไม่ได้แล้ว ต้องลบตัวอักษรเก่าออกเพื่อสร้างเนื้อที่ ให้ตัวอักษรใหม่ที่เป็นต้องสร้าง) การลบตัวอักษรที่ผู้ใช้สร้างต้องเรียกใช้จาก โปรแกรมย่อย SYMBOL โดยกำหนดตัวแปร NCHAR เท่ากับ -1 หรือ เท่ากับ -2 (โปรดดูโปรแกรมย่อย SYMBOL)

รูปแบบการเรียกใช้โปรแกรมย่อย USRSYM

```

INTEGER * 4 ISYM
INTEGER * 4 IOFSET (n)
(n จำนวนเวกเตอร์ที่ใช้สำหรับตัวอักษรตัวนี้)
INTEGER * 4 N
CALL USRSYM (ISYM, IOFSET, N)

```

ความหมายของตัวแปรต่างๆ

ตัวแปร ISYM เป็นตัวกำหนดรหัสของตัวอักษรที่จะสร้างขึ้นนี้และค่านี้ใช้ ตอนเรียกใช้โปรแกรมย่อย SYMBOL

ISYM มีค่าระหว่าง 288 กับ 383 สำหรับชุดตัวอักษรที่ 1

ISYM มีค่าระหว่าง 383 กับ 479 สำหรับชุดตัวอักษรที่ 2

ตัวแปร IOFSET เป็นชุดข้อมูลที่เก็บเวกเตอร์ของตัวอักษรที่ผู้ใช้กำหนด
 ล้างการเก็บเวกเตอร์ใน IOFSET จะเก็บค่า VX, VY โดยอยู่ในรูป
 $VX*100+VY$ และค่า VX และ VY มีค่ามากที่สุด 14 น้อยที่สุด 0
 ถ้า VX มีค่าเท่ากับ 15 และ VY มีค่าเท่ากับ 0 ค่าที่เก็บไว้นี้
 หมายความว่าให้ย้ายตำแหน่งต่อไปโดยไม่มีการลากเส้น (ยกปากกาขึ้น
 ตอนย้ายตำแหน่ง)

ถ้า VX มีค่าเท่ากับ 15 และ VY มีค่าเท่ากับ 15 ด้วยหมายความว่า
 ให้ย้ายตำแหน่งมายังตำแหน่งตรงกลาง (7,7) โดยไม่มีการลากเส้น
 ตัวแปร N ใช้บอกจำนวน เวกเตอร์ที่จะส่งมาในชุดข้อมูล IOFSET
 แต่ถ้า N มีค่าเท่ากับ -1001 จะลบตัวอักษรชุดที่ 1 ออกหมด
 แต่ถ้า N มีค่าเท่ากับ -1002 จะลบตัวอักษรชุดที่ 2 ออกหมด

3.3.11 โปรแกรมย่อย NUMBER

ใช้แสดงผลลัพธ์ของจำนวนตัวเลขออกมาเป็นตัวอักษร โดยกำหนดจำนวนหลักหลังจุดทศนิยมได้ด้วย ใช้ในเวลาที่ต้องการพลอตค่าตัวเลขต่างๆ ออกมาสำหรับจำนวนเต็มก็สามารถแสดงผลได้ โดยแปลงเป็นจำนวนจริงก่อน พร้อมกับกำหนดจำนวนหลักหลังจุดทศนิยมว่าไม่ต้องการรวมทั้งจุดทศนิยมด้วย

รูปแบบการเรียกใช้โปรแกรมย่อย NUMBER

```
REAL*4      X, Y, HITE, FPN, ANGLE
INTEGER*4    NDEC
CALL NUMBER (X, Y, H, FPN, ANGLE, NDEC)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X, Y ใช้กำหนดตำแหน่งเริ่มต้นที่จะใช้พลอตตัวเลข

ถ้า X มีค่าเท่ากับ 999. จะใช้ค่า X ของตำแหน่งล่าสุดที่มีการพลอต

ถ้า Y มีค่าเท่ากับ 999. จะใช้ค่า Y ของตำแหน่งล่าสุดที่มีการพลอต

ตัวแปร HITE ใช้กำหนดความสูงของตัวอักษรที่เป็นตัวเลขของตัวอักษรที่เป็นตัวเลขที่จะทำการพลอตแสดงออกมา

ตัวแปร FPN คือค่าจำนวนจริงที่เก็บค่าตัวเลขที่ต้องแสดงผลลัพธ์ออกมา

ตัวแปร ANGLE ใช้กำหนดมุมเอียงของแถวตัวเลขที่ทำการพลอต

ตัวแปร NDEC ถ้ามีค่าเท่ากับศูนย์ จะพลอตแต่ส่วนที่เป็นจำนวนเต็มเท่านั้นพร้อมกับมีจุดทศนิยมปิดท้าย และมีการปัดเศษ(round)

ถ้า NDEC มีค่าเท่ากับ -1 จะทำการแสดงผลลัพธ์เฉพาะส่วนที่เป็นจำนวนเต็มหลังการปัดเศษแล้ว ไม่มีจุดทศนิยมปิดท้าย

ถ้า NDEC มีค่าน้อยกว่า -1 จะหมายความว่ากำหนดจำนวนหลักหลังจุดทศนิยมว่าจะมีตัวเลขกี่ตัวและตัดเศษทิ้งเลย(truncate) โดยจำนวนหลักเท่ากับ ค่าสัมบูรณ์ของ NDEC ลบด้วยค่า 1 ($|NDEC| - 1$)

3.3.12 โปรแกรมย่อย SCALE

ใช้ในกรณีที่ผู้ใช้ที่ต้องการพลอตกราฟเส้น โดยปกติแล้วค่าของข้อมูลที่จะทำการพลอตกราฟจะไม่พอดีกับความยาวของแกนกราฟที่ใช้ เพราะฉะนั้นต้องมีอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาว เพื่อใช้คำนวณหาตำแหน่งที่จะทำการพลอตจะได้มีค่าตรงกับค่ากำกับแกนกราฟ นอกจากนี้ยังต้องหาค่าที่น้อยที่สุดของข้อมูลในชุดข้อมูลซึ่งจะเป็นค่าเริ่มต้นที่ใช้กำกับตามแกนของกราฟ เช่น ค่าที่น้อยที่สุดคือ 1 และค่าที่มากที่สุดคือ 11 และความยาวแกนคือ 5 หน่วย เพราะฉะนั้นค่าที่ใช้กำกับตอนเริ่มต้นแกนคือ 1 และ อัตราส่วนค่าของค่าของข้อมูล (ซึ่งใช้คำนวณค่ากำกับแกนกราฟด้วย) คือ ค่าที่มากที่สุดลบด้วยค่าที่น้อยที่สุดแล้วหารด้วยความยาวของแกนกราฟ $((11 - 1) / 5 = 2)$ ซึ่งจะได้ค่ากำกับแกนกราฟดังนี้ คือ 1, 2, 3, 5, 7, 9, 11

รูปแบบการเรียกใช้โปรแกรมย่อย SCALE

```
REAL*4      ARRAY (n)
```

(ค่า n นี้จะมีค่าเท่ากับจำนวนข้อมูลทั้งหมดที่จะทำการพลอตกราฟเส้นคุณกับค่าที่กำหนดว่าจะใช้ข้อมูลตัวไหน (INC) ในชุดข้อมูล พร้อมกับเพิ่มอีก

2 คุณกับ ค่าที่กำหนดเลือกข้อมูล (INC)

(สรุปคือ $(NPTS + 2) * INC$)

```
REAL*4      AXISLENG
```

```
INTEGER*4   NPTS, INC
```

```
CALL SCALE (ARRAY, AXISLENG, NPTS, INC)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร ARRAY ใช้เก็บข้อมูลที่จะทำพลอตกราฟ และล้ารองที่ว่างสำหรับให้โปรแกรมย่อย SCALE คำนวณค่าส่งกลับมา 2 ค่าคือค่าที่น้อยที่สุดของข้อมูลในชุดข้อมูล และอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาวของแกนกราฟ

ตัวแปร AXISLENG ใช้บอกความยาวของแกนกราฟที่ใช้

ตัวแปร NPTS ใช้บอกจำนวนของจุดข้อมูลที่จะทำการพลอต อย่างน้อยต้องมีค่าเท่ากับ 2

ตัวแปร INC ใช้บอกระยะห่างระหว่างข้อมูลในชุดข้อมูล สำหรับเลือกข้อมูลในชุดข้อมูลมาทำการพลอต เช่น ถ้า $INC = 2$ จะใช้ข้อมูลตัวที่ 2 ข้อมูลตัวที่ 4 ข้อมูลตัวที่ 6 และต่อไปจนครบจำนวนข้อมูลที่กำหนดมา (NPTS) หรือถ้า $INC = 1$ จะใช้ข้อมูลตัวที่ 1 ข้อมูลตัวที่ 2 ข้อมูลตัวที่ 3 ข้อมูลตัวที่ 4 และต่อไปจนครบจำนวนข้อมูลที่กำหนด ฯลฯ ค่าเริ่มต้นจะเป็นค่าที่มีค่าน้อยที่สุด และอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาวของแกนกราฟ จะเก็บไว้ตอนท้ายของชุดข้อมูล คือที่

$ARRAY(NPTS*INC+1)$ และ $ARRAY(NPTS*INC+INC+1)$

ตามลำดับ โปรแกรมย่อย SCALE เหมาะสำหรับกรณีที่ค่าของข้อมูลในชุดข้อมูลที่มีความแตกต่างกันมากกับความยาวแกนกราฟมาก

3.3.13 โปรแกรมย่อย AXIS

ใช้ในการพลอตแกนกราฟ เพื่อให้อ้างอิงในการอ่านค่าของกราฟที่พลอต โดยสามารถกำหนดความยาวแกนกราฟ มุมเอียงของแกนกราฟ ข้อความที่ใช้อธิบายแกนกราฟตรงกลางของแกนกราฟ

ข้อแนะนำ โดยปกติการพลอตกราฟต้องมีแกน 2 แกน คือแกนแนวนอน (แกน-X) และ แกนแนวตั้ง (แกน-Y) ต้องเรียกใช้โปรแกรมย่อยนี้ 2 ครั้ง ครั้งแรกเรียกโดยมีมุมเอียง 0 องศา สำหรับแกนแนวนอน(แนว-X) ครั้งที่สองเรียกโดยมีมุมเอียง 90 องศา สำหรับแกนแนวตั้ง(แกน-Y)

รูปแบบการเรียกโปรแกรมย่อย AXIS

```
REAL*4      X, Y, AXISLENG, FIRSTV, DELTAV, ANGLE
INTEGER*4   NCHAR
CHARACTER*n CHRSTRING
(n คือจำนวนตัวอักษรที่จะใช้สำหรับข้อความกำกับแกนกราฟ)
CALL  AXIS (X, Y, CHRSTRING, NCHAR, AXISLENG, ANGLE,
*        FIRSTV, DELTAV)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X, Y ใช้กำหนดตำแหน่งเริ่มต้นของแกน ควรห่างจากขอบของจอภาพประมาณ 1. หน่วย ให้มีที่พอสำหรับข้อความอธิบายแกนกราฟ

ตัวแปร CHRSTRING ใช้เก็บข้อความที่จะใช้อธิบายแกนกราฟ

ตัวแปร NCHAR ใช้เก็บจำนวนตัวอักษรของข้อความอธิบายแกนกราฟ

ตัวแปร AXISLENG ใช้กำหนดความยาวของแกนกราฟ

ตัวแปร FIRSTV ใช้กำหนดค่าเริ่มต้นสำหรับกำกับแกนกราฟ

ตัวแปร DELTAV ใช้กำหนดอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาว

ตัวแปร ANGLE ใช้กำหนดมุมเอียงของแกนกราฟที่จะพลอต

3.3.14 โปรแกรมย่อย LINE

ใช้สำหรับพลอตกราฟของชุดข้อมูลสองชุด (ชุดหนึ่งสำหรับแกนแนวราบ และ อีกชุดหนึ่งสำหรับแกนแนวตั้ง) โดยต้องส่งค่าที่ใช้เริ่มต้น (ค่าที่น้อยที่สุดที่จะปรากฏบนแกนของกราฟ) โดยค่านี้ตอนท้ายของชุดข้อมูล คือที่

```
ARRAY (NPTS * INC + 1)
```

และต้องส่งอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาว ซึ่งใช้สำหรับเพื่อใช้คำนวณ ตำแหน่งที่จะทำการพลอตซึ่งแทนค่าของข้อมูล จะได้ตรงกับค่ากำกับแกนกราฟ โดยค่านี้จะอยู่ที่

```
ARRAY (NPTS * INC + INC + 1)
```

ข้อแนะนำ การพลอตกราฟจุดควรใช้ในกรณีข้อมูลมีความหนาแน่นไม่มากนัก มิฉะนั้นตัวอักษรที่ถูกลบออกจากตำแหน่งซึ่งแทนค่าของข้อมูล อาจจะทำให้รูปภาพผิดปกติ

รูปแบบการเรียกใช้โปรแกรมย่อย LINE

```
REAL*4      XARRAY (n), YARRAY(n)
```

```
INTEGER*4   NPTS, INC, LINTYP, CHRTYP
```

(ค่า n นี้จะมีค่าเท่ากับ $NPTS * INC + INC + 1$ เพื่อให้ที่สำหรับ เก็บค่าเริ่มต้น และค่าเพิ่มสำหรับแต่ละช่วงของแกนกราฟ)

```
CALL LINE (XARRAY, YARRAY, NPTS, INC, LINTYP, CHRTYP)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร XARRAY ใช้เก็บข้อมูลของแกนแนวตั้ง (แกน-X) และรวม ค่าที่น้อยที่สุดซึ่งใช้เป็นค่าเริ่มต้นบนแกนแนวตั้ง และอัตราส่วนค่าของ ข้อมูลต่อหนึ่งหน่วยความยาว

ตัวแปร YARRAY ใช้เก็บข้อมูลของแกนแนวดิ่ง (แกน-Y) และรวมค่าที่น้อยที่สุดซึ่งใช้เป็นค่าเริ่มต้นบนแกนแนวดิ่ง และอัตราส่วนค่าของข้อมูลต่อหนึ่งหน่วยความยาว

ตัวแปร NPTS ใช้กำหนดจำนวนข้อมูลที่ใช้พลอตกราฟ โดยจำนวนของข้อมูลแกนแนวนอนต้องเท่ากับ จำนวนข้อมูลแกนแนวดิ่ง

ตัวแปร INC กำหนดระยะห่างระหว่างข้อมูลในชุดข้อมูล ใช้เลือกข้อมูลในชุดข้อมูลมาทำการพลอตเช่น ถ้า $INC = 2$ จะใช้ข้อมูลตัวที่ 2 ข้อมูลตัวที่ 4 ข้อมูลตัวที่ 6 และต่อไปจนครบจำนวนข้อมูลที่กำหนดมา หรือถ้า $INC = 1$ จะใช้ข้อมูลตัวที่ 1 ข้อมูลตัวที่ 2 ข้อมูลตัวที่ 3 ข้อมูลตัวที่ 4 และต่อไปจนครบจำนวนข้อมูลที่กำหนด (NPTS) ฯลฯ

ตัวแปร LINTYP ใช้กำหนดลักษณะของเส้นกราฟที่ใช้พลอตดังต่อไปนี้ ถ้า $LINTYP = 0$ จะมีเส้นทึบเชื่อมระหว่างจุดข้อมูล และไม่มีตัวอักษรในแต่ละจุดข้อมูล เหมาะสำหรับการใช้พลอตกราฟเส้นของชุดข้อมูลที่มีปริมาณหนาแน่นมากต่อความยาวแกนที่มีอยู่

ถ้า $LINTYP > 0$ จะมีได้ 9 ค่าจาก 1 ถึง 9 โดยดูได้ว่าลักษณะเส้นได้จากโปรแกรมย่อย DASHLN และแต่ละตำแหน่งที่แทนค่าของข้อมูล จะมีตัวอักษรอยู่ด้วย และสามารถนำเส้นพลอตกราฟเส้นไปเปรียบเทียบกันได้ โดยกำหนดลักษณะของเส้นกราฟให้ต่างกันและกำหนดตัวอักษรที่ใช้ต่างกัน ถ้า $LINTYP < 0$ กราฟที่ได้จะเป็นกราฟจุดของตัวอักษรเท่านั้น

ตัวแปร CHRTYP ใช้กำหนดตัวอักษรที่จะใช้พลอตที่ตำแหน่งที่แทนค่าของข้อมูล ค่าที่ใช้ได้มีค่าตั้งแต่ 0-9

(โปรดดูตัวอักษรพิเศษที่ใช้พลอตกราฟได้จากภาคผนวก ก. ในตัวอย่างผลลัพธ์ที่ 3)

3.3.15 โปรแกรมย่อย CIRCLE

ใช้สำหรับพลอตรูปวงกลม หรือส่วนหนึ่งของวงกลม โดยมีรูปแบบที่รับเข้า
ได้ 7 แบบ ตามที่แสดงในรูปที่ 3-2

Function	P1	P2	P3	P4	P5	IMODE
Full Circle	Xs	Ys	Xc	Yc	dummy	1
Arc(minor)	Xs	Ys	Xe	Ye	R	2
Arc(major)	Xs	Ys	Xe	Ye	R	3
Full Circle	Xs	Ys	As	dummy	R	4
Arc	Xs	Ys	As	Ae	R	5
Full Circle	Xc	Yc	As	dummy	R	6
Arc	Xc	Yc	As	Ae	R	7

รูปที่ 3-2

รูปแบบการเรียกใช้โปรแกรมย่อย CIRCLE

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

INTEGER*4 IMODE

ความหมายของตัวแปรต่างๆ

ตัวแปร IMODE ใช้กำหนดความหมายของพารามิเตอร์ที่รับเข้ามาตามรูปที่ 3-2

ตัวแปร P1, P2, P3, P4, P5 เป็นพารามิเตอร์ที่ถูกกำหนดความหมายตามค่าของตัวแปร IMODE ซึ่งจะกำหนดตามรูปที่ 3-2

- (X_s, Y_s คือ จุดเริ่มต้นที่จะทำการพลอตวงกลม หรือส่วนของวงกลม)
- (X_e, Y_e คือ จุดสิ้นสุดที่จะทำการพลอตวงกลม หรือส่วนของวงกลม)
- (X_c, Y_c คือ จุดศูนย์กลางของวงกลม)
- (R คือ รัศมีของวงกลม)
- (A_s คือ มุมเริ่มต้นที่ทำการพลอต อ้างอิงกับแกนแนวราบ ซึ่งเท่ากับ ศูนย์องศา)
- (A_e คือ มุมสิ้นสุดที่ทำการพลอต อ้างอิงกับแกนแนวราบ ซึ่งเท่ากับ ศูนย์องศา)
- (major คือ ส่วนของวงกลมที่ยาวมากกว่า จากจุดเริ่มต้น ไปยังจุดสิ้นสุด)
- (minor คือ ส่วนของวงกลมที่ยาวน้อยกว่า จากจุดเริ่มต้น ไปยังจุดสิ้นสุด)

3.3.16 โปรแกรมย่อย OPMES (Operation MESSAGES)

ใช้สำหรับส่งข้อความมายังจอภาพระหว่างทำการพลอตภาพ เพื่อแสดงข้อความตามผู้ใช้ต้อง เช่น ในกรณีที่โปรแกรมทำงานไปนานๆ ผู้ใช้ต้องการทราบว่าโปรแกรมทำงานไปถึงขั้นตอนไหนแล้ว

รูปแบบการเรียกใช้โปรแกรมย่อย OPMES

INTEGER*4 NCT

CHARACTER*n XMSG

(n เป็นจำนวนตัวอักษร ตั้งแต่ความต้องการของผู้ใช้ว่าต้องการแสดงข้อความกี่ตัวอักษร)

CALL OPMES (NCT, XMSG)

ความหมายของตัวแปรต่าง ๆ

ตัวแปร NCT ใช้บอกจำนวนตัวอักษรของข้อความที่ต้องการแสดงบนจอภาพ

ถ้า $NCT > 0$ จอภาพจะมีแต่ข้อความที่แสดงเท่านั้น

ถ้า $NCT < 0$ จอภาพจะมีข้อความที่แสดง และ ภาพที่ทำการพลอตด้วย

ตัวแปร XMSG ใช้เก็บข้อความที่ผู้ใช้ต้องการแสดงออกระหว่างการพลอตภาพ

3.4 โปรแกรมย่อยทางกราฟิกที่มีหน้าที่พิเศษเฉพาะอย่าง

3.4.1 โปรแกรมย่อย RPLOT

ใช้สำหรับเปลี่ยนตำแหน่งการพลอต ซึ่งอ้างอิงกับตำแหน่งพลอตล่าสุด ไม่อ้างอิงกับจุดกำเนิดเหมือนโปรแกรมย่อย PLOT เช่น ถ้าจุดสุดท้ายที่มีการพลอตคือ (2,3) ถ้ามีเรียกโปรแกรมย่อย RPLOT (4,5,2) หมายความว่ามีการพลอตจากตำแหน่ง (2,3) ไปยังตำแหน่ง (6,8) ฯลฯ

รูปแบบการเรียกใช้โปรแกรมย่อย RPLOT

```
REAL*4      X,Y
INTEGER*4   IPEN
CALL RPLOT (X,Y,IPEN)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X,Y กำหนดตำแหน่งที่จะทำการพลอต (ในระบบหน่วยความยาวของผู้ใช้) โดยตำแหน่งนี้จะอ้างอิงกับตำแหน่งล่าสุดที่มีการพลอต ตัวแปร IPEN กำหนดว่าการเปลี่ยนตำแหน่งการพลอตนี้ว่ามีการลากเส้นด้วยหรือไม่ ถ้ามีค่าเป็นลบหมายความว่าให้ตำแหน่งที่กำหนดมาจะถูกกำหนดเป็นจุดกำเนิดใหม่ใช้อ้างอิงตำแหน่งต่อไป ส่วนถ้าค่าเป็นบวกจะไม่มีการกำหนดจุดกำเนิดใหม่ ให้ใช้จุดกำเนิดที่มีอยู่เดิมต่อไป และค่าลิมบอร์นของ IPEN นี้จะมีเพียง 3 ค่า คือ

IPEN มีค่าเท่ากับ 2 หมายความว่า การย้ายตำแหน่งนี้ให้มีการลากเส้นจากจุดล่าสุดมายังจุดที่กำหนดมาด้วย

IPEN มีค่าเท่ากับ 3 หมายความว่า มีแต่การย้ายตำแหน่งอย่างเดียว

IPEN มีค่าเท่ากับ 999 หมายความว่า มีการพลอตรูปได้สิ้นสุดลงแล้ว จะเลิกการเรียกใช้โปรแกรมย่อยทางกราฟิก

3.4.2 โปรแกรมย่อย RELPLT

ใช้สำหรับทำการพลอต แทนการเรียกโปรแกรมย่อย RPLLOT หรือ โปรแกรมย่อย PLOT ในกรณีที่ผู้ใช้ต้องการให้มีการเอียงภาพ โดยการใช้การเรียกใช้ โปรแกรมย่อย ROTATE กำหนดมุมเอียงภาพ และ ตำแหน่งที่อ้างอิงในการพลอต จะอ้างอิงจากตำแหน่งล่าสุดที่มีการพลอต เหมือนกับโปรแกรมย่อย RPLLOT

รูปแบบการเรียกใช้โปรแกรมย่อย RELPLT

```
REAL*4      X, Y
INTEGER*4   IPEN
CALL RELPLT (X, Y, IPEN)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X, Y กำหนดตำแหน่งที่จะทำการพลอต (ในระบบหน่วยความยาวของผู้ใช้) โดยตำแหน่งนี้จะมีการหมุนให้เอียงตามมุมที่กำหนดจากโปรแกรมย่อย ROTATE และ อ้างอิงกับตำแหน่งล่าสุดที่มีการพลอต

ตัวแปร IPEN กำหนดว่าการเปลี่ยนตำแหน่งการพลอตนี้มีการลากเส้นด้วยหรือไม่ ถ้ามีค่าเป็นลบหมายความว่าให้ตำแหน่งที่กำหนดมาจะถูกกำหนดเป็นจุดกำเนิดใหม่ให้อ้างอิงตำแหน่งต่อไป ส่วนถ้าค่าเป็นบวกจะไม่มีกำหนดจุดกำเนิดใหม่ ให้ใช้จุดกำเนิดที่มีอยู่เดิมต่อไป และค่าลัมบ์ของ IPEN นี้จะมีเพียง 3 ค่า คือ

IPEN มีค่าเท่ากับ 2 หมายความว่า การย้ายตำแหน่งนี้ให้มีการลากเส้นจากจุดล่าสุดมายังจุดที่กำหนดมาด้วย

IPEN มีค่าเท่ากับ 3 หมายความว่า มีแต่การย้ายตำแหน่งอย่างเดียว

IPEN มีค่าเท่ากับ 999 หมายความว่า มีการพลอตรูปได้สิ้นสุดลงแล้ว จะเลิกการเรียกใช้โปรแกรมย่อยทางกราฟิก

3.4.3 โปรแกรมย่อย DASHLN (DASH LiNe)

ใช้สำหรับลากเส้นในลักษณะต่าง ๆ เช่น เส้นทึบ เส้นจุดไข่ปลา ที่มีระยะห่างระหว่างจุดไข่ปลาหลายอย่าง เส้นประที่มีความยาวต่างกัน ฯลฯ

รูปแบบการเรียกใช้โปรแกรมย่อย DASHLN

REAL*4 X1,Y1,X2,Y2

INTEGER*4 MODE

ความหมายของตัวแปรต่าง ๆ

ตัวแปร	X1, Y1	กำหนดจุดเริ่มต้นของเส้นตรง
ตัวแปร	X2, Y2	กำหนดจุดสิ้นสุดของเส้นตรง
ตัวแปร	MODE	กำหนดลักษณะของเส้นที่ต้องการ
ถ้า	MODE = 0	คือเส้นทึบ
	= 1	คือเส้นจุดไข่ปลา 50 จุดต่อหนึ่งหน่วยความยาว
	= 2	คือเส้นจุดไข่ปลา 25 จุดต่อหนึ่งหน่วยความยาว
	= 3	คือเส้นจุดไข่ปลา 10 จุดต่อหนึ่งหน่วยความยาว
	= 4	คือเส้นจุดไข่ปลา 5 จุดต่อหนึ่งหน่วยความยาว
	= 5	คือเส้นประ รอยประมีความยาว 1/16 หน่วย
	= 6	คือเส้นประ รอยประมีความยาว 1/8 หน่วย
	= 7	คือเส้นประ รอยประมีความยาว 1/4 หน่วย
	= 8	คือเส้นประ รอยประมีความยาว 1/2 หน่วย
	= 9	คือเส้นประ รอยประมีความยาว 1 หน่วย

3.4.4 โปรแกรมย่อย AROHD (ARrOw HeAd)

ใช้สำหรับพลอตรูปลูกศร โดยมีลักษณะของหัวลูกศรต่างๆกันตามแต่ความต้องการของผู้ใช้ เช่น ใช้สำหรับเขียนบอกขนาดความยาวหรือใช้ใช้อธิบายข้อความกับรูปภาพ ฯลฯ โดยผู้ใช้ ต้องกำหนดจุดเริ่มต้น และจุดปลายหัวลูกศร และขนาดของหัวลูกศร และแบบของลูกศร

รูปแบบการเรียกใช้โปรแกรมย่อย AROHD

```
REAL*4 ITYPE
```

```
CALL AROHD (X,Y,XTIP,YTIP,AHLEN,AHWD,ITYPE)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X,Y กำหนดตำแหน่งเริ่มต้นของลูกศร

ตัวแปร XTIP,YTIP กำหนด ตำแหน่งปลายหัวลูกศร

ตัวแปร AHLEN กำหนดขนาดความยาวของหัวลูกศร

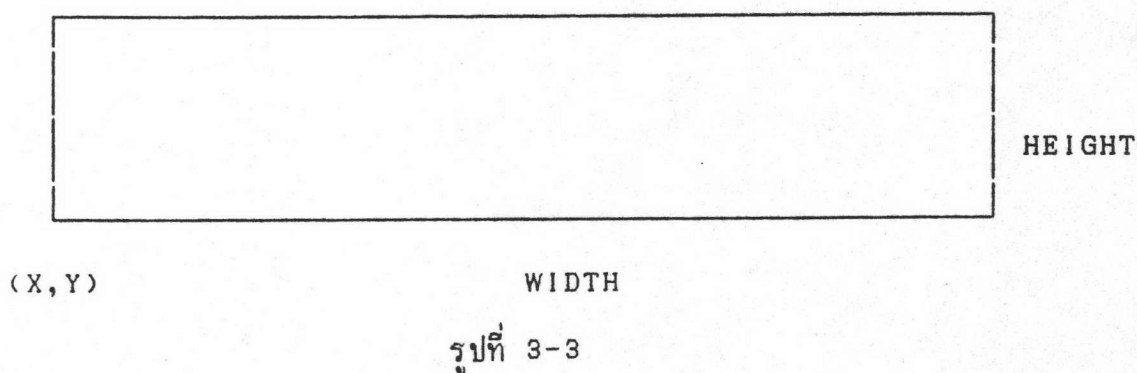
ตัวแปร AHWD กำหนดขนาดความกว้างของหัวลูกศร

ตัวแปร ITYPE กำหนดแบบของหัวลูกศร

ถ้า ITYPE = 1	หัวลูกศรจะเป็นดังนี้	---	>
ถ้า ITYPE = 2	หัวลูกศรจะเป็นดังนี้	---	>
ถ้า ITYPE = 3	หัวลูกศรจะเป็นดังนี้	---	>

3.4.5 โปรแกรมย่อย RCTNGL (ReCTaNGLe)

ใช้สำหรับพลอตรูปสี่เหลี่ยมขนาดต่างๆ ตามความต้องการของผู้ใช้ โดยผู้ใช้งานกำหนดมาคือ ตำแหน่งของมุมซ้ายล่าง และความกว้างและความยาวของรูปสี่เหลี่ยม ดังเช่นรูปที่ 3-3



รูปแบบการเรียกใช้โปรแกรมย่อย RCTNGL

```
REAL*4      X, Y, HEIGHT, WIDTH, THETA
INTEGER*4   IZ
CALL RCTNGL (X, Y, HEIGHT, WIDTH, THETA, IZ)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X, Y	กำหนดตำแหน่งล่างซ้ายของรูปสี่เหลี่ยม
ตัวแปร H	กำหนดความสูงของรูปสี่เหลี่ยม
ตัวแปร W	กำหนดความกว้างของรูปสี่เหลี่ยม
ตัวแปร THETA	กำหนดมุมของรูปสี่เหลี่ยมดังรูปที่ 3-3
ตัวแปร IZ	กำหนดว่ามีการลากเส้นจากตำแหน่งสุดท้ายที่มีการพลอตมายังตำแหน่งมุมล่างซ้าย
ถ้า IZ = 2	มีการลากเส้น
ถ้า IZ = 3	ไม่มีการลากเส้น

3.4.6 โปรแกรมย่อย RCTBLK (ReCTaNGLe BLock)

ใช้สำหรับพลอตรูปสี่เหลี่ยมทึบขนาดต่างๆ ตามความต้องการของผู้ใช้ กำหนดมาคือ ตำแหน่งของมุมล่างซ้าย และ ความกว้างและความยาวของ รูปสี่เหลี่ยมทึบเหมือนโปรแกรมย่อย RCTNGL

รูปแบบการเรียกใช้โปรแกรมย่อย RCTNGL

```
REAL*4 X,Y,H,W,THETA
INTEGER*4 IZ
CALL RCTBLK (X,Y,H,W,THETA,IZ)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร	X, Y	กำหนดตำแหน่งล่างซ้ายของรูปสี่เหลี่ยมทึบ
ตัวแปร	H	กำหนดความสูงของรูปสี่เหลี่ยมทึบ
ตัวแปร	W	กำหนดความกว้างของรูปสี่เหลี่ยมทึบ
ตัวแปร	THETA	กำหนดมุมของรูปสี่เหลี่ยมทึบ
ตัวแปร	IZ	กำหนดว่ามีการลากเส้นจากตำแหน่งสุดท้ายที่มีการพลอตมายังตำแหน่งมุมล่างซ้าย
ถ้า	IZ = 2	มีการลากเส้น
ถ้า	IZ = 3	ไม่มีการลากเส้น

3.4.7 โปรแกรมย่อย BAR

ใช้สำหรับพลอตกราฟแท่งที่ข้อมูลมีจำนวนไม่มากนักประมาณ 2-20 ข้อมูล จะทำให้สามารถวิเคราะห์ผลลัพธ์ได้ง่าย โดยเพียงแต่ผู้ใช้ส่งแต่ชุดข้อมูลกับจำนวน ข้อมูลให้โปรแกรมย่อยเท่านั้น (โปรดดูตัวอย่างโปรแกรมในภาคผนวก ก. จะแสดง วิธีเขียนโปรแกรมสร้างกราฟแท่ง)

รูปแบบการเรียกใช้โปรแกรมย่อย BAR

```
REAL*4 X, Y, XLENG, ARRAY
```

```
INTEGER*4 NBAR
```

```
CALL BAR (X, Y, ARRAY, NBAR, XLENG)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร	X, Y	กำหนดจุดมุมล่างซ้ายของกราฟแท่ง
ตัวแปร	ARRAY	เก็บชุดข้อมูลของผู้ใช้
ตัวแปร	XLENG	กำหนดความยาวของแกนแนวนอน (แกน X)
ตัวแปร	NBAR	กำหนดจำนวนข้อมูลที่ไปพลอตกราฟ

3.4.8 โปรแกรมย่อย BAR2

ใช้สำหรับพลอตกราฟแท่งเปรียบเทียบระหว่างข้อมูล 2 ชุดเปรียบเทียบกัน จำนวนข้อมูลควรจะมีประมาณ 2-20 ข้อมูล ช่วยให้สามารถวิเคราะห์ผลลัพธ์ได้ง่าย โดยเพียงแต่ผู้ใช้ ส่งข้อมูลมา 2 ชุด และจำนวนข้อมูลที่จะทำการเปรียบเทียบกัน โดยจำนวนข้อมูลของชุดข้อมูลต้องเท่ากัน (โปรดดูตัวอย่างการเรียกใช้โปรแกรมย่อยนี้ ในภาคผนวก ก. ซึ่งจะแสดงวิธีการเขียนโปรแกรมสร้างกราฟแท่งเปรียบเทียบ)

รูปแบบการเรียกใช้โปรแกรมย่อย BAR2

```
REAL*4 X,Y,XLENG,ARRAY1, ARRAY2
INTEGER*4 NBAR
CALL BAR2 (X,Y,ARRAY1,ARRAY2,NBAR,LENG)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร X,Y กำหนดตำแหน่งล่างซ้ายของกราฟแท่งเปรียบเทียบ

ตัวแปร ARRAY1,ARRAY2 เก็บชุดข้อมูลทั้งสองชุดที่ใช้พลอตกราฟแท่งเปรียบเทียบ

ตัวแปร XLENG กำหนดความยาวของแกนแนวนอน (แกน-X)

ตัวแปร NBAR กำหนดจำนวนข้อมูลที่จะพลอตเป็นแท่งกราฟเปรียบเทียบ

3.4.9 โปรแกรมย่อย PIE

ใช้สำหรับพลอตกราฟรูปร่างวงกลม เป็นการเปรียบเทียบอัตราส่วนค่าของข้อมูลแต่ละข้อมูลกับค่ารวมทั้งหมดของข้อมูล จำนวนที่ใช้ควรประมาณ 2-20 ข้อมูล เนื่องจากความละเอียดของอุปกรณ์แสดงผลมีจำกัด ถ้าข้อมูลมากไปกราฟจะไม่สามารถอ่านค่าได้ (โปรดดูตัวอย่างการเรียกใช้โปรแกรมย่อยนี้ในภาคผนวก ก. ซึ่งจะแสดงวิธีเขียนโปรแกรมสร้างกราฟวงกลม)

รูปแบบการเรียกใช้โปรแกรมย่อย PIE

```
REAL*4      XC, YC, R, ARRAY
INTEGER*4   NPIE, MODE
CALL PIE (XC, YC, ARRAY, NPIE, R, MODE)
```

ความหมายของตัวแปรต่าง ๆ

ตัวแปร	XC, YC	กำหนดจุดศูนย์กลางของกราฟวงกลม
ตัวแปร	R	กำหนดรัศมีของวงกลม
ตัวแปร	ARRAY	เก็บข้อมูลที่ทำการพลอตกราฟวงกลม
ตัวแปร	NPIE	กำหนดจำนวนข้อมูลที่ใช้พลอตกราฟวงกลม
ตัวแปร	MODE = 1	พลอตกราฟวงกลมรวม
ตัวแปร	MODE = 2	พลอตวงกลมแต่ละเส้นแยกจากกัน