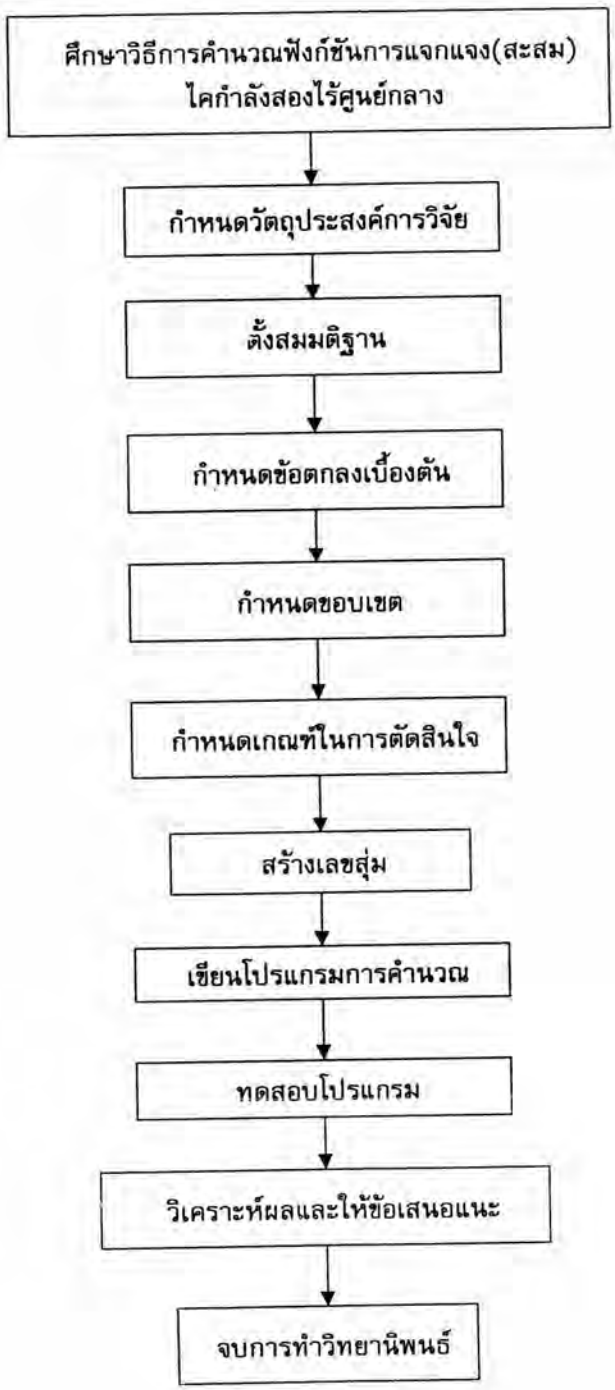
บทที่ 3
วิธีดำเนินการวิจัย

3.1 ขั้นตอนการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงวิธีการดำเนินงานวิจัยของวิธีการคำนวณฟังก์ชันการแจกแจงสะสมไค-กำลังสองไร้ศูนย์กลางโดยวิธีการของแพ็ตเนค (วิธีที่ 1) วิธีการคำนวณของแอชอาวร์และแอ็บเคล-ซาแมคในกรณีองศาความเป็นอิสระเป็นค่าใด ๆ(วิธีที่ 2) วิธีการคำนวณของแอชอาวร์และแอ็บเคล-ซาแมคในกรณีองศาความเป็นอิสระเป็นเลขคี่ (วิธีที่ 3) และวิธีคำนวณของรูเบ็น เบิก และโกวิน-ดาราจูลู (วิธีที่ 4) โดยขั้นตอนการดำเนินงานวิจัยประกอบด้วย การศึกษาวิธีการคำนวณฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง การกำหนดวัตถุประสงค์ การตั้งสมมติฐาน การกำหนดข้อตกลงเบื้องต้น การกำหนดขอบเขต การกำหนดเกณฑ์ในการตัดสินใจ การสร้างเลขสุ่ม การเขียนโปรแกรม การทดสอบโปรแกรมที่เขียน ตลอดจนการวิเคราะห์ผลและให้ข้อเสนอแนะ สามารถแสดงผังงานการดำเนินงานได้ดังนี้ โดยที่รูปภาพแสดงขั้นตอนการดำเนินงานวิจัยจะแสดงในรูปภาพที่ 3.1

ขั้นตอนการดำเนินงานวิจัย



รูปภาพที่ 3.1

3.2 เครื่องมือที่ใช้ในการดำเนินงานวิจัย

3.2.1.ภาษาที่ใช้ในการดำเนินงานวิจัย การจำลองแบบด้วยเทคนิคการจำลองแบบและการผลิตเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอรวมถึงการการคำนวณค่าฟังก์ชันการแจกแจงสะสมไค-กำลังสองไร้ศูนย์กลางถูกพัฒนาด้วยภาษาซี (C language) ซึ่งเป็นภาษาระดับที่ 3

3.2.2.เครื่องคอมพิวเตอร์ที่ใช้ในการวิจัย เครื่องคอมพิวเตอร์ที่ใช้เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล (personal computer หรือ PC) รุ่น 486 DX2-66 , หน่วยความจำ(RAM) 20 MB

3.2.3. เทคนิคการจำลองแบบ (Simulation Technique)

การจำลองแบบเป็นเทคนิคการทดลองพื้นฐานที่มีความรวดเร็วและเสียค่าใช้จ่ายถูกกว่าในการทดลองจริง เรานำการจำลองแบบมาใช้ในเหตุการณ์ดังต่อไปนี้

1. การแก้ปัญหาในทางปฏิบัติของการศึกษาทดลอง เช่น การศึกษาทางด้านอุตุนิยมวิทยาด้านการจราจรของรถยนต์และเครื่องบินต้องใช้การคำนวณตามหลักคณิตศาสตร์ที่ซับซ้อนมากหรืออาจจะไม่สามารถแก้ปัญหาได้เลย

2. การเกิดปัญหาที่มีลักษณะไม่แน่นอน หรือ ปัญหาที่มีลักษณะเคลื่อนไหวแบบไดนามิก (dynamic)ระหว่างโอกาสการเกิดตามลำดับเหตุการณ์ย่อยหรือความสัมพันธ์ต่อกันที่ซับซ้อนระหว่างตัวแปรในระบบซึ่งแต่ละแบบของเหตุการณ์ที่กล่าวมานี้มีโอกาสเกิดขึ้นเองได้แล้วแต่สภาพการณ์และเป็นการเสียค่าใช้จ่ายที่สูงมาก และยังเสียเวลานานมากในการทำการทดลองจริง

3. การจำลองแบบจะสามารถกระทำขึ้นโดยเสียค่าใช้จ่ายที่ถูกกว่าและรวดเร็วกว่าพร้อมทั้งสามารถลดช่วงว่างระหว่างการทดลองจริงกับในทางปฏิบัติ แม้ในบางครั้งปัญหานี้จะสามารถวิเคราะห์ได้โดยสูตรทางคณิตศาสตร์ แต่การจำลองแบบจะช่วยลดเวลาในการคำนวณอย่างมาก

3.2.3.1.การผลิตเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอ (Uniformly distributed random number)

การผลิตเลขสุ่มโดยการใช้โปรแกรม

"เป็นการผลิตเลขสุ่มจากความสัมพันธ์เวียนบังเกิด(recurrence relation) กล่าวคือเลขตัวถัดไปเกิดจากการดำเนินการทางคณิตศาสตร์และตรรกศาสตร์ ด้วยตัวเลขปัจจุบันหรือกลุ่มของตัวเลขในอดีต อนุบรรพของเลขซึ่งผลิตในลักษณะนี้จึงไม่ใช่อนุบรรพของเลขสุ่มที่แท้จริง อนุบรรพของเลขซึ่งผลิตในลักษณะนี้ย่อมมีคาบ แต่อย่างไรก็ตามเลขที่ผลิตออกมาเหล่านี้อาจผ่านการทดสอบความเป็นสุ่มเชิงสถิติหลายอย่างได้ จึงเรียกเลขเหล่านี้ว่าเลขคล้ายสุ่ม(pseudo-random number)

การผลิตเลขสุ่มโดยการใช้โปรแกรมมีข้อดีหลายประการ ที่สำคัญคือวิธีนี้สามารถผลิตอนุบรรพเลขสุ่มชุดเดิมออกมาได้ ซึ่งเป็นสิ่งจำเป็นอย่างยิ่งในกรณีที่ทดสอบแบบจำลองและมีความประสงค์ที่จะทบทวนการคำนวณโปรแกรมสำหรับเลขสุ่มมีความง่ายและสั้นมาก

ข้อบกพร่องของวิธีการผลิตนี้ก็คือ อนุบรรพของเลขสุ่มที่ผลิตออกมาเป็นอนุบรรพที่มีคาบทั้งสิ้น และการอนุมานคุณสมบัติทางสถิติของเลขสุ่มเหล่านี้ทางทฤษฎีทำได้ยากพอสมควร โปรแกรมที่ใช้ผลิตเลขสุ่มควรมีคุณสมบัติพื้นฐานดังต่อไปนี้

1. เลขที่ผลิตด้วยการใช้โปรแกรมจะต้องมีสหสัมพันธ์กันน้อย

2.การแจกแจงของเลขที่ผลิตจากการใช้โปรแกรมจะต้องใกล้เคียงกับการแจกแจงมาตรฐานมากที่สุดเท่าที่จะมากได้

3.โปรแกรมที่ใช้คำนวณเลขสุ่มจะต้องมีเสถียรภาพกล่าวคือการแจกแจงของเลขจะต้องไม่เปลี่ยนแปลงไปตามจำนวนตัวเลขที่ผลิตออกมา

เทคนิคการผลิตเลขสุ่มโดยการใช้โปรแกรมได้รับการพัฒนาอย่างรวดเร็ว ฟอนนอยมาน (Vonneuman) และ เมโทรโปลิส(Metropolis) ได้เสนอวิธีตัวกลางกำลังสอง(mid-square method) ในปีค.ศ.1946 ฟอร์โซธ(Forsythe) ได้ปรับปรุงวิธีตัวกลางกำลังสอง และต่อมาเลเมอร์(Lehmer) ได้เสนอวิธีผลิตเลขสุ่มด้วยการใช้เศษจากการหารผลคูณ(multiplicative congruential method) ซึ่งเป็นวิธีที่ใช้กันอย่างแพร่หลายในคอมพิวเตอร์ ยุคปัจจุบันต่อมาโรเทนลีร์(Rotenleerg) ได้ปรับปรุงวิธีการของเลเมอร์เป็นการใช้เศษของผลบวกของผลคูณกับค่าคงที่จากการหาร(mixed congruential method)"[1]

<u>วิธีการผลิตเลขสุ่มด้วยการใช้เศษจากการหารผลคูณ (Multiplicative Congruential Method)</u>

วิธีนี้จะหาเลขสุ่มโดยทำการคำนวณจากสมการที่ (3.1)

(3.1) $x_{i+1} = x_i \cdot a \text{ (MODULO M)}$

เมื่อ x_i เป็นเลขคล้ายสุ่มตัวที่ i

x_{i+1} เป็นเลขคล้ายสุ่มตัวที่ i+1

และ a เป็นตัวคูณคงที่ (constant multiplier)

---

[1] สมเกียรติ เกตุเอี่ยม ," การเปรียบเทียบวิธีการประมาณค่าพารามิเตอร์เพื่อการทดสอบการแจกแจงของประชากรที่ให้ค่าสถิติไคสแควร์ต่ำสุด " (วิทยานิพนธ์ปริญญามหาบัณฑิตสาขาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2528) , หน้า 27.

MODULO M หมายความว่า ค่า x_i .a ถูกหารด้วย M จนกระทั่งเหลือเศษน้อยกว่า M เลขที่เหลือเศษจึงเป็นเลขคล้ายสุ่มตัวต่อไปคือ x_{i+1}

วิธีการเริ่มต้นโดยค่าเริ่มต้น x เรียกวาค่าเริ่มต้น (initial value) ตัวอย่างการผลิตเลขสุ่มโดยวิธีการนี้เช่นกำหนดให้ค่าเริ่มต้น x = 3 , a = 7 และ M =15 ดังนั้นกลุ่มเลขคล้ายสุ่มจะได้ดังนี้

$x_1 = 3 \times 7 = 21 \equiv 6$

$x_2 = 6 \times 7 = 42 \equiv 12$

$x_3 = 12 \times 7 = 84 \equiv 9$

$x_4 = 9 \times 7 = 63 \equiv 3$

เช่นนี้เรื่อยไปจะได้เลขคล้ายสุ่มที่มีค่าอยู่ระหว่าง 0-14

จากการใช้สมการ(3.1) เราจะเห็นได้ว่าเลขคล้ายสุ่มที่ได้เป็นเลขจำนวนเต็มเค่าใดค่าหนึ่งในช่วง 0, 1, 2, ..., M-1 หลังจากนั้นแล้วจะได้เลขสุ่มชุดเดิมอีก ฉะนั้นคาบของเลขคล้ายสุ่มที่ได้จึงมีค่าไม่เกิน M (แต่อาจจะน้อยกว่า M ก็ได้ เมื่อเลือกค่า a และ x ไม่ดีนัก) การเลือกค่า M a และ x จึงมีความสำคัญในการผลิตเลขคล้ายสุ่มที่มีคาบใกล้เคียงกับค่า M มากที่สุด

การทดลองเลือกใช้ค่า M a และ x ที่จับคู่ต่าง ๆกันเพื่อใช้ผลิตเลขคล้ายสุ่มตามสมการที่ (3.1) พบว่าถ้าเลือก x เป็นเลขคี่ และ $M = 2^r$ (เมื่อ $r > 2$) และ $a = K . 8 \pm 3$ (เมื่อ K เป็นเลขจำนวนเต็มบวกใด ๆ)จะได้คาบของเลขคล้ายสุ่มมากที่สุดและเท่ากับ 2^{r-2} วิธีการต่อไปนี้เป็นวิธีการเลือกค่าพารามิเตอร์ทั้ง 3 ตัว เพื่อจะได้กลุ่มของเลขคล้ายสุ่มที่ดี

1.เลือกค่า M (M >1) ค่า M จะต้องเป็นค่าจำนวนเต็มที่ใหญ่ที่สุดที่สามารถคำนวณได้จากเครื่องคอมพิวเตอร์ ในการศึกษาครั้งนี้ใช้ภาษาซี ซึ่งสามารถเก็บค่าจำนวนเต็มแต่ละค่าได้เท่ากับ 32 บิต ซึ่งบิตสุดท้าย 1 บิตใช้สำหรับแสดงเครื่องหมาย ดังนั้นเลขจำนวนเต็มที่ใหญ่ที่สุดใน 1 ค่า ที่คอมพิวเตอร์รับได้คือ $2^{31}-1 = 2147483647$ นั่นคือค่า M ควรมีค่า 2147483648

2. ค่าเริ่มต้น x ควรเป็นเลขจำนวนเฉพาะ กับค่า M เมื่อ M เป็นค่ากำลังสอง(จาก $M=2^r$) ดังนั้น x จึงควรมีค่าเป็นเลขจำนวนเต็มบวกที่เลขคี่(ในกรณีที่ใช้ x เป็นเลขคู่จะพบว่าทุก ๆค่า x_i ตัวต่อไปจะเป็นเลขคู่เสมอ จึงไม่มีคุณสมบัติเลขสุ่ม)

3. ค่าคงที่ที่ใช้เป็นตัวคูณ a (constant multiplier) ควรมีค่าเป็นจำนวนเฉพาะ กับค่า M ด้วย นั่นคือ a ต้องเป็นเลขคี่ ในที่นี้ให้ a = 16807

การสร้างเลขสุ่ม ประกอบด้วยโปรแกรม 3 ส่วน คือ

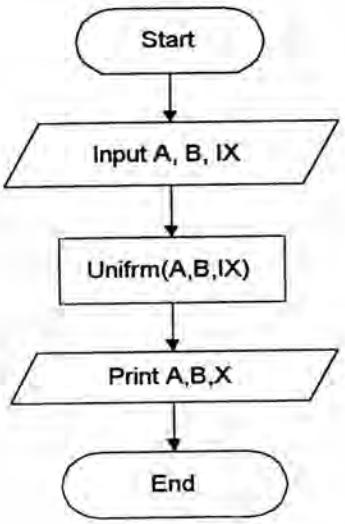
1. โปรแกรมหลัก **(Main Program)** โปรแกรมนี้เป็นโปรแกรมซึ่งเรียกโปรแกรมย่อย unifrm ผังงานโปรแกรมหลักแสดงได้ดังรูปภาพที่ 3.2

2. โปรแกรมย่อย unifrm โปรแกรมนี้เป็นโปรแกรมซึ่งทำหน้าที่สร้างเลขสุ่มที่มีการแจกแจงสม่ำเสมอ ผังงานโปรแกรมย่อย unifrm แสดงได้ดังรูปภาพที่ 3.3

3. โปรแกรมย่อย randu โปรแกรมนี้เป็นโปรแกรมซึ่งทำหน้าที่สร้างเลขสุ่มให้กับโปรแกรมย่อย unifrm ผังงานโปรแกรมย่อย randu แสดงได้ดังรูปภาพที่ 3.4

โปรแกรมหลัก (Main Program) โปรแกรมหลักนี้จะรับค่าเริ่มต้น ค่าสิ้นสุดของการแจกแจงสม่ำเสมอ และรับค่าเริ่มต้นที่ใช้ในการสร้างเลขสุ่ม โปรแกรมนี้เรียกใช้งานโปรแกรมย่อย unifrm ผังงานโปรแกรมหลักแสดงได้ดังรูปภาพที่ 3.2

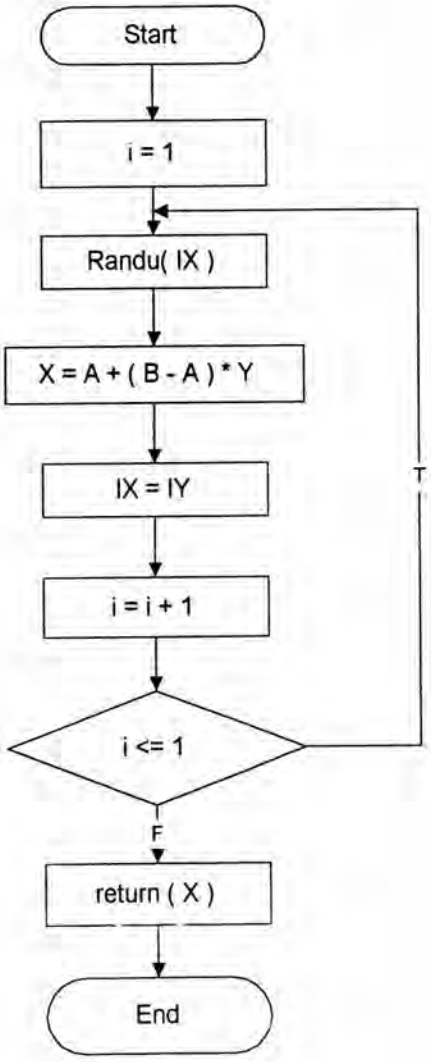
โปรแกรมหลัก (Main Program)



รูปภาพที่ 3.2

โปรแกรมย่อย unifrm โปรแกรมนี้ทำหน้าที่สร้างเลขสุ่มที่มีการแจกแจงสม่ำเสมอ และโปรแกรมนี้เรียกใช้งานโปรแกรมย่อย randu ผังงานโปรแกรมย่อย unifrm แสดงได้ดังรูปภาพที่ 3.3

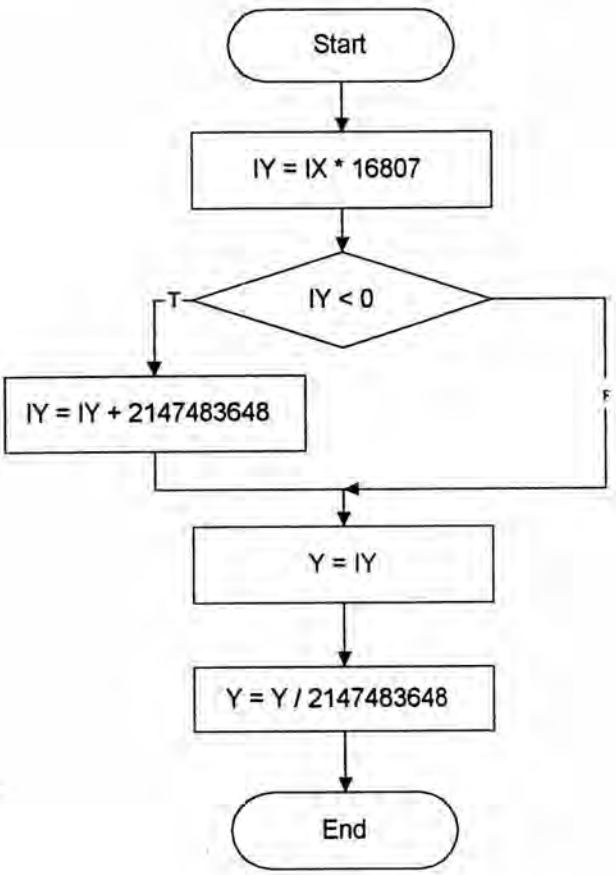
<u>โปรแกรมย่อย unifrm</u>



รูปภาพที่ 3.3

โปรแกรมย่อย randu เป็นโปรแกรมที่ทำหน้าที่สร้างเลขสุ่มให้กับโปรแกรมย่อย unifrm ผังงานโปรแกรมย่อย randu แสดงได้ดังรูปภาพที่ 3.4

<u>โปรแกรมย่อย randu</u>



รูปภาพที่ 3.4

3.2.4. ส่วนประกอบของโปรแกรมและฟังก์ชันที่ใช้ในการดำเนินงานวิจัย

โปรแกรมหลัก(Main Program)ที่ใช้ในการดำเนินงานวิจัยประกอบด้วยโปรแกรมย่อยดังนี้

1. โปรแกรมย่อย patnaik โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของแพ็ตเนคทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางเมื่อองศาความเป็นอิสระเป็นเลขจำนวนเต็มใด ๆ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน 2 ฟังก์ชันคือ ฟังก์ชัน gammds และ ฟังก์ชัน chi2

2. โปรแกรมย่อย anyash โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของแอชอาวร์และแอ็บเดล-ซาแมด ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางเมื่อองศาความเป็นอิสระเป็นเลขจำนวนเต็มใด ๆ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน 3 ฟังก์ชัน คือ ฟังก์ชัน alngam ฟังก์ชัน gamma และ ฟังก์ชัน chi1

3. โปรแกรมย่อย oddash โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของแอชอาวร์และแอ็บเดล-ซาแมด ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง เมื่อองศาความเป็นอิสระเป็นเลขจำนวนเต็มคี่ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน gammds

4. โปรแกรมย่อย ruben โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของรูเบ็น เบิก และ โกวินดาราจูลู ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง เมื่อองศาความเป็นอิสระเป็นเลขจำนวนเต็มคี่ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน alnorm

ฟังก์ชันที่โปรแกรมย่อยเรียกใช้งานมีดังนี้

1. ฟังก์ชัน alnorm

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าการแจกแจงปกติสะสมซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย ruben

2. ฟังก์ชัน alngam

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าล็อกกาลิทึมของฟังก์ชันการแจกแจงแกมมา ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik โปรแกรมย่อย anyash และโปรแกรมย่อย oddash

3. ฟังก์ชัน gammds

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงแกมมาโดยที่ฟังก์ชันนี้เรียกใช้งานฟังก์ชัน alngam ด้วย ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik และโปรแกรมย่อย oddash

4. ฟังก์ชัน gamma

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงแกมมา ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย anyash

5. ฟังก์ชัน chi1

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงไคกำลังสอง ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย anyash

6. ฟังก์ชัน chi2

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงไคกำลังสอง ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik

ตารางแสดงการเรียกใช้งานฟังก์ชันของโปรแกรมย่อย แสดงดังตารางที่ 3.1

ตารางที่ 3.1

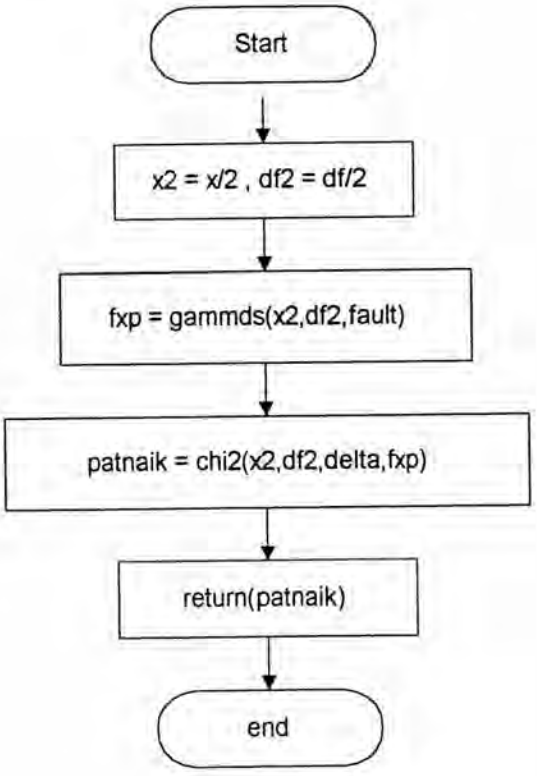
ตารางแสดงการเรียกใช้งานฟังก์ชันของโปรแกรมย่อย

โปรแกรมย่อย	ฟังก์ชัน					
	alnorm	alngam	gammds	gamma	chi1	chi2
patnaik		x	x			x
anyash		x		x	x	
oddash		x	x			
ruben	x					

โปรแกรมหลัก ที่ใช้ในการดำเนินงานวิจัยประกอบด้วยโปรแกรมย่อย 4 โปรแกรมด้วยกัน คือ โปรแกรมย่อย patnaik โปรแกรมย่อย anyash โปรแกรมย่อย oddash และโปรแกรมย่อย ruben อาจ แสดงได้ดังนี้

1. โปรแกรมย่อย patnaik โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของแพ็ตเนคทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง เมื่อองศาความเป็นอิสระเป็นเลขจำนวนเต็มใด ๆ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน 2 ฟังก์ชันคือ ฟังก์ชัน gammds และฟังก์ชัน chi2 ผังงานโปรแกรมย่อย patnaik แสดงได้ดังรูปภาพที่ 3.5

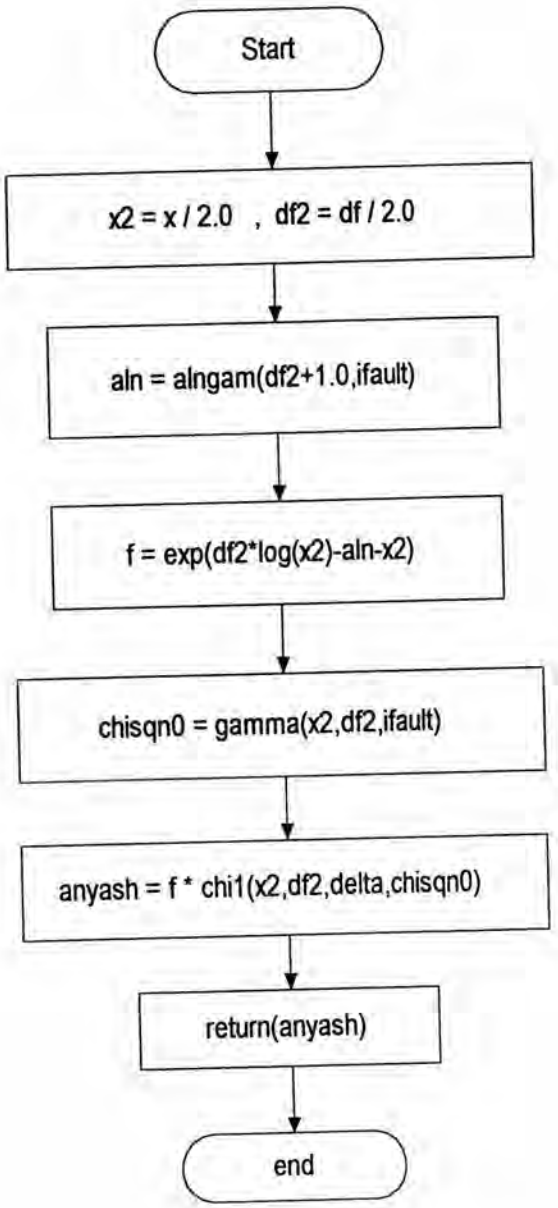
<u>โปรแกรมย่อย patnaik</u>



รูปภาพที่ 3.5

2. โปรแกรมย่อย anyash โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของแอชอาวร์และแอ็บเดล-ซาแมด ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง เมื่อองศาความเป็นอิสระ เป็นเลขจำนวนเต็มใด ๆ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน 3 ฟังก์ชันคือ ฟังก์ชัน alngam ฟังก์ชัน gamma และฟังก์ชัน chi1 ผังงานโปรแกรมย่อย anyash แสดงได้ดังรูปภาพที่ 3.6

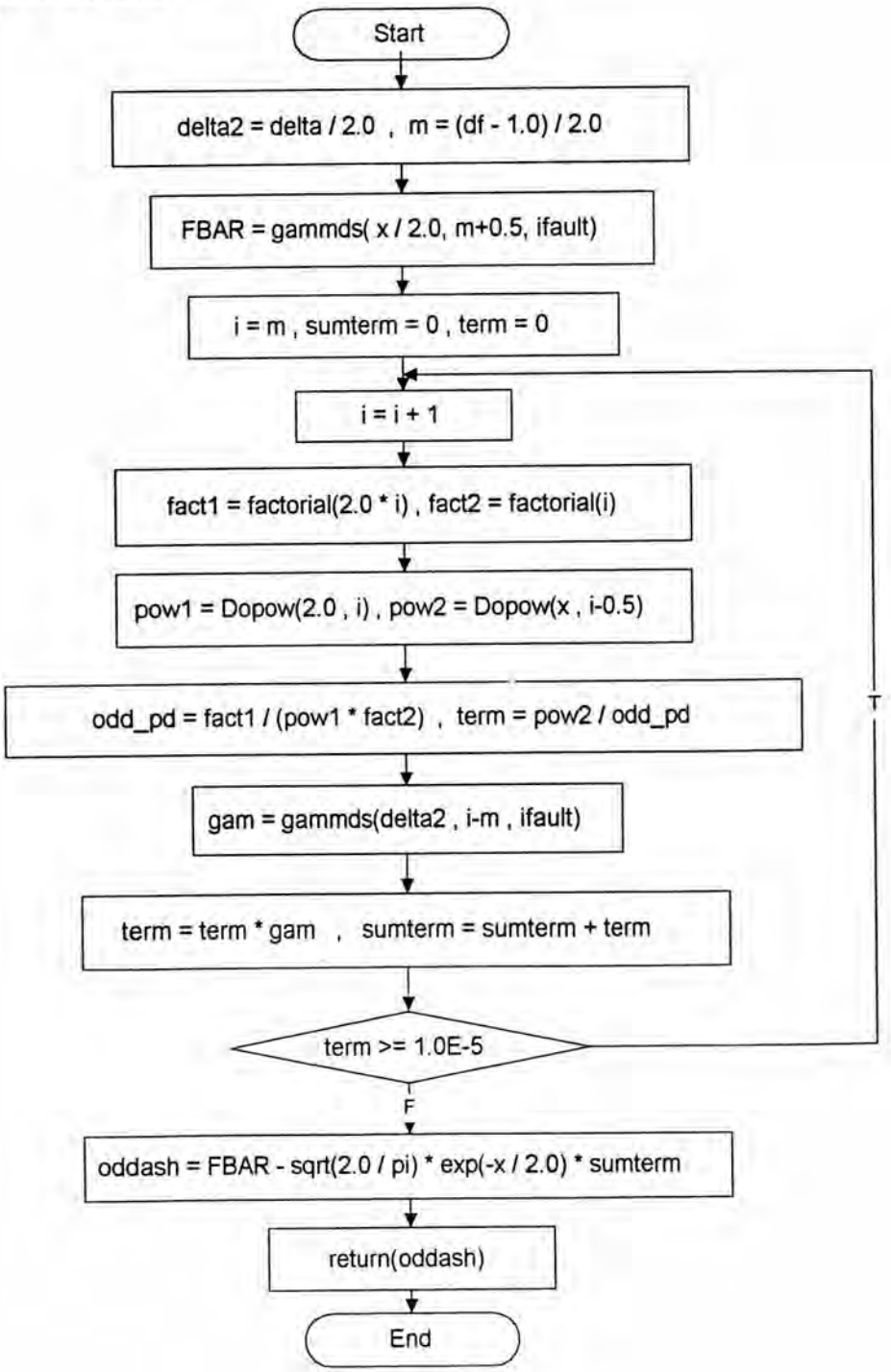
<u>โปรแกรมย่อย anyash</u>



รูปภาพที่ 3.6

3. โปรแกรมย่อย oddash โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของแอชอาวร์และแอ็บเดล-ซาแมด ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลาง เมื่อองศาความเป็นอิสระเป็นเลขจำนวนเต็มคี่ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน gammds ผังงานโปรแกรมย่อย oddash แสดงได้ดังรูปภาพที่ 3.7

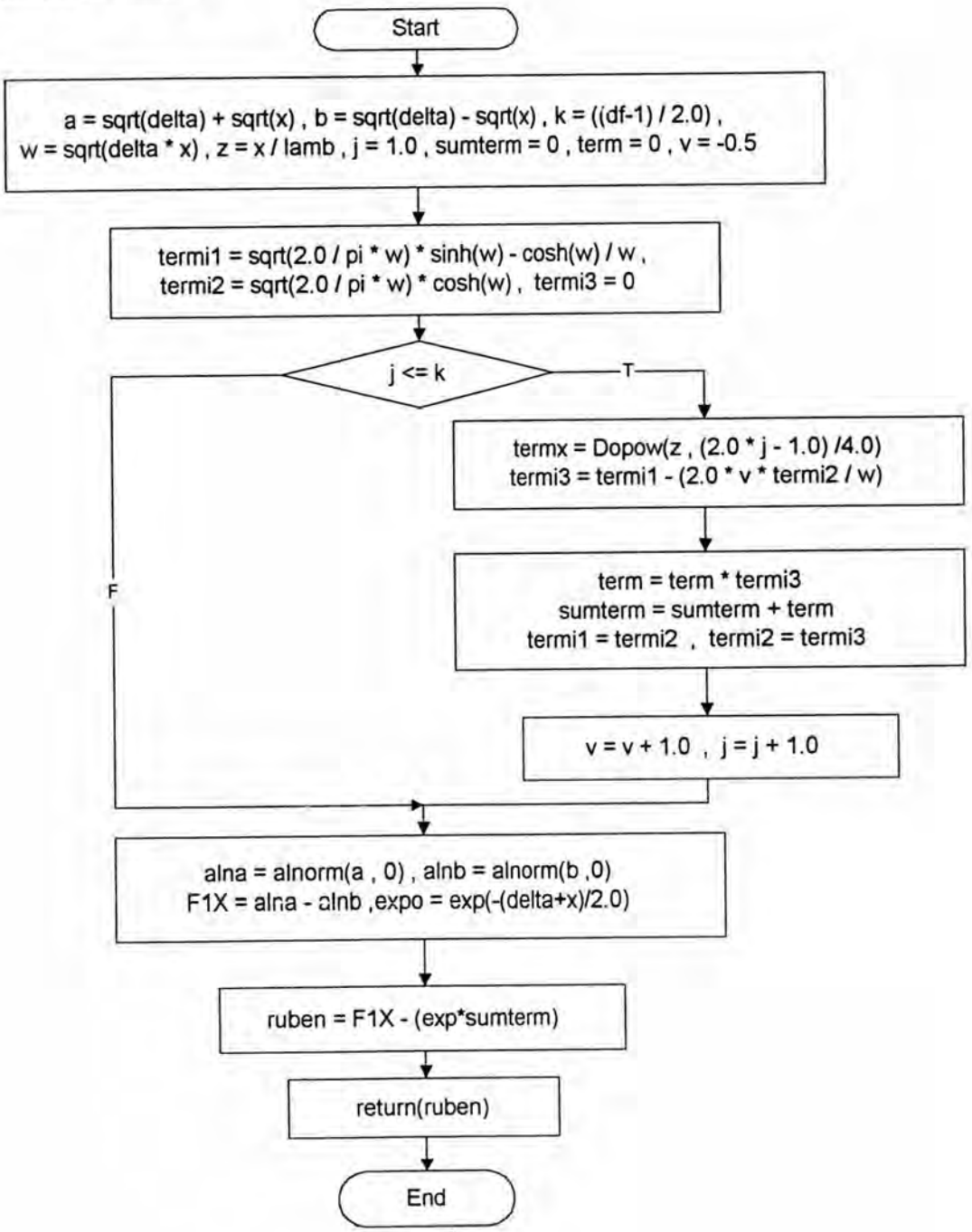
<u>โปรแกรมย่อย oddash</u>



รูปภาพที่ 3.7

4. โปรแกรมย่อย ruben โปรแกรมย่อยนี้เป็นโปรแกรมที่ใช้ในการคำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางโดยวิธีการคำนวณของ รูเบ็น เบิก และโกวินดาราจูลู ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงสะสมไคกำลังสองไร้ศูนย์กลางเมื่อองศาความเป็นอิสระเป็นเลขจำนวนเต็มคี่ และโปรแกรมย่อยนี้เรียกใช้งานฟังก์ชัน alnorm ผังงานโปรแกรมย่อย ruben แสดงได้ดังรูปภาพที่ 3.8

โปรแกรมย่อย ruben



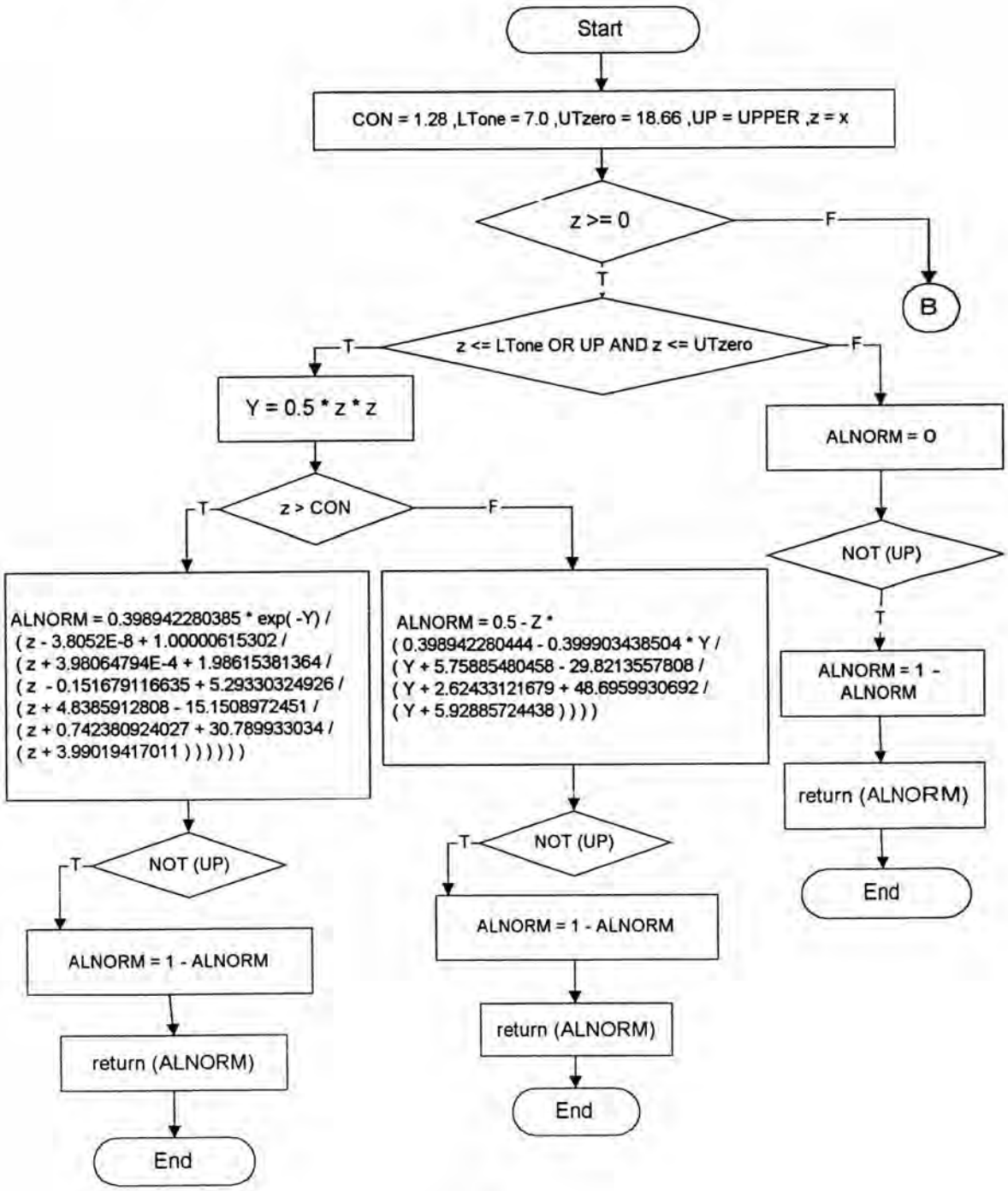
รูปภาพที่ 3.8

ฟังก์ชันที่ใช้ในการคำนวณ

1. ฟังก์ชัน alnorm

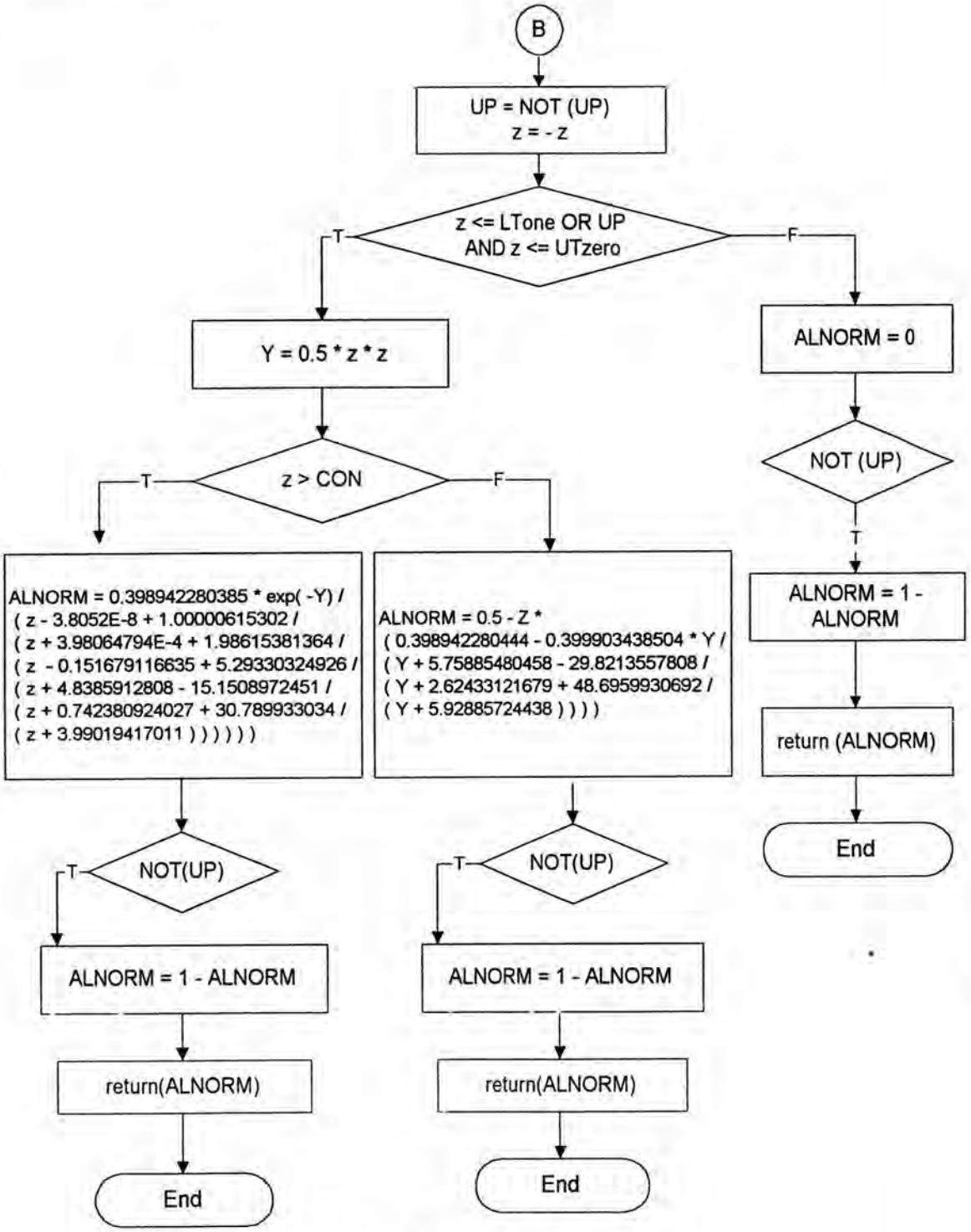
ฟังก์ชันนี้ทำหน้าที่คำนวณค่าการแจกแจงปกติสะสม ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย ruben และผังงานฟังก์ชัน alnorm ซึ่งแสดงได้ดังรูปภาพที่ 3.9

ฟังก์ชัน alnorm



รูปภาพที่ 3.9

ฟังก์ชัน alnorm (ต่อ)

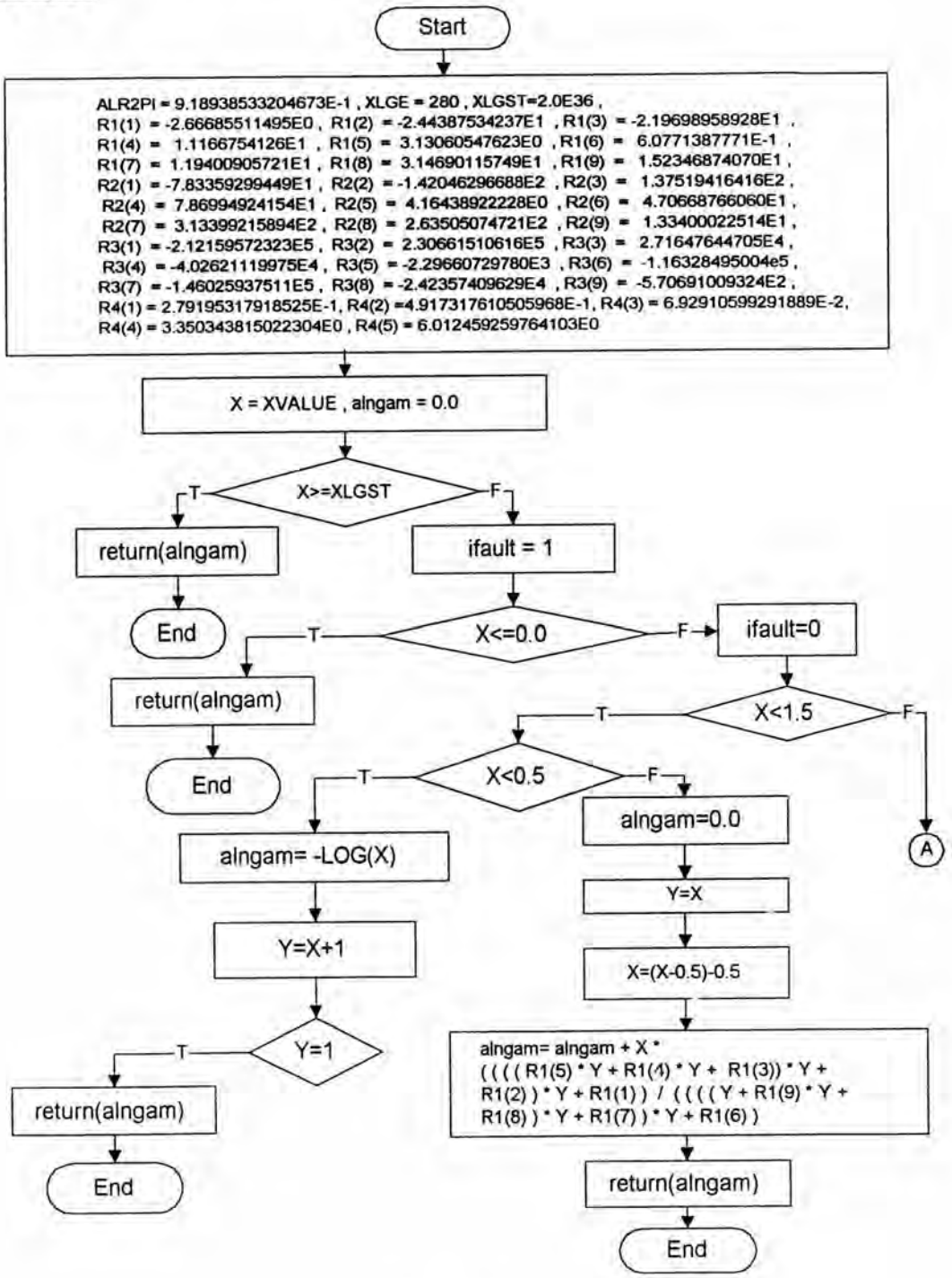


รูปภาพที่ 3.9 (ต่อ)

2. ฟังก์ชัน alngam

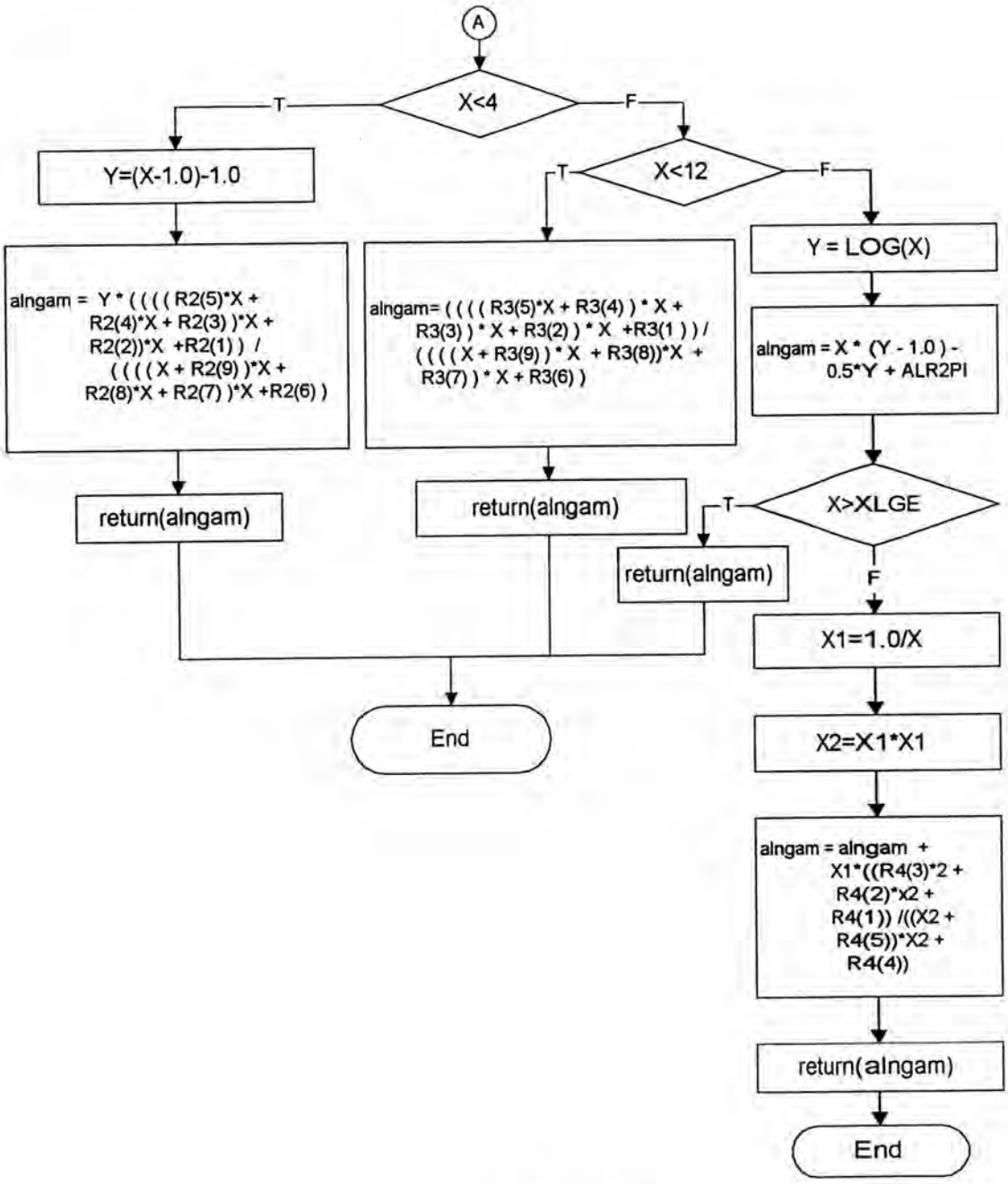
ฟังก์ชันนี้ทำหน้าที่คำนวณค่าล็อกกาลิทึมของฟังก์ชันการแจกแจงแกมมา ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik โปรแกรมย่อย anyash และโปรแกรมย่อย oddash และผังงานฟังก์ชัน alngam ซึ่งแสดงได้ดังรูปภาพที่ 3.10

ฟังก์ชัน alngam



รูปภาพที่ 3.10

ฟังก์ชัน alngam (ต่อ)

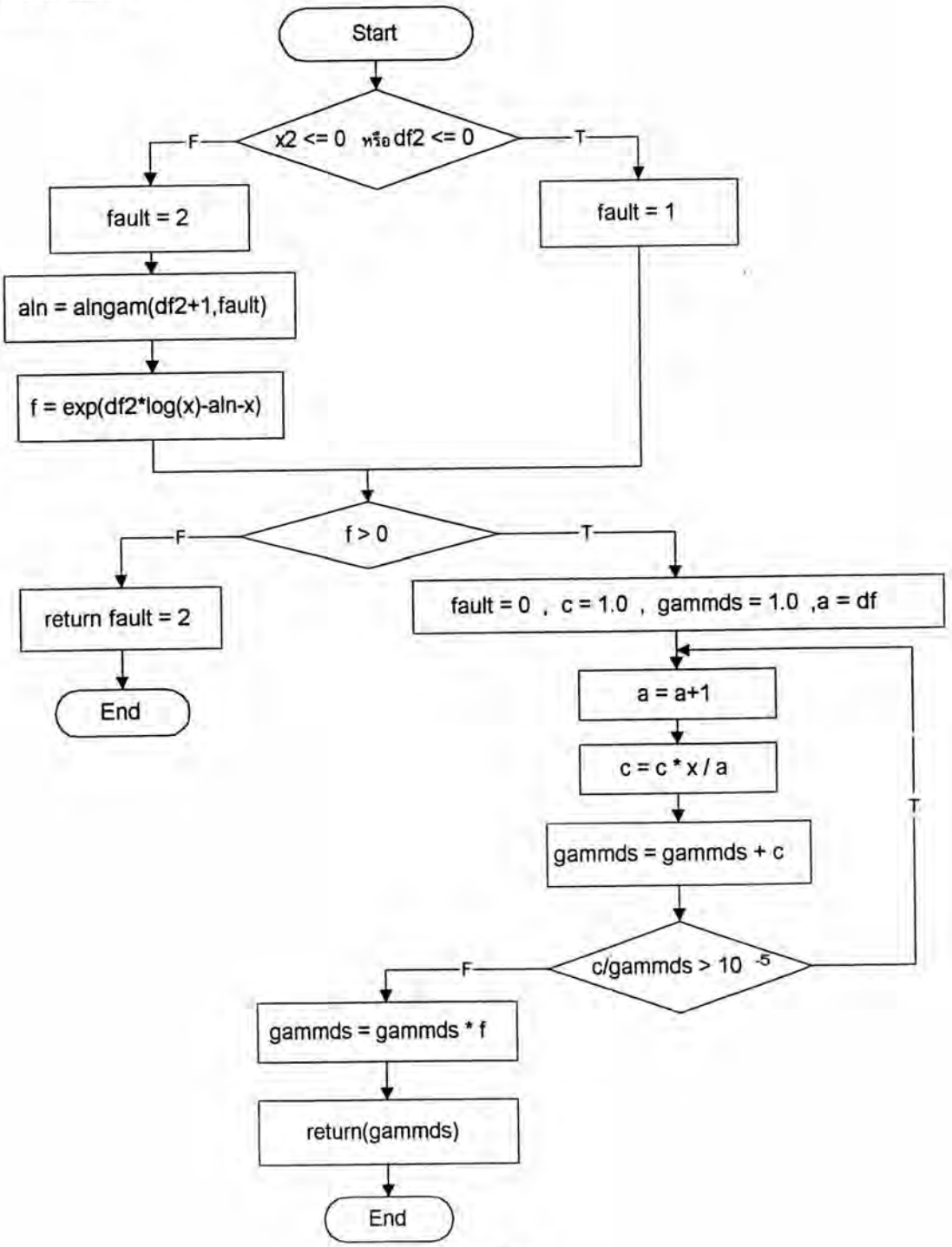


รูปภาพที่ 3.10(ต่อ)

3. ฟังก์ชัน gammds

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงแกมมา โดยที่ฟังก์ชันนี้เรียกใช้งานฟังก์ชัน alngam ด้วย ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย patnaik และโปรแกรมย่อย oddash และผังงานฟังก์ชัน gammds ซึ่งแสดงได้ดังรูปภาพที่ 3.11

<u>ฟังก์ชัน gammds</u>

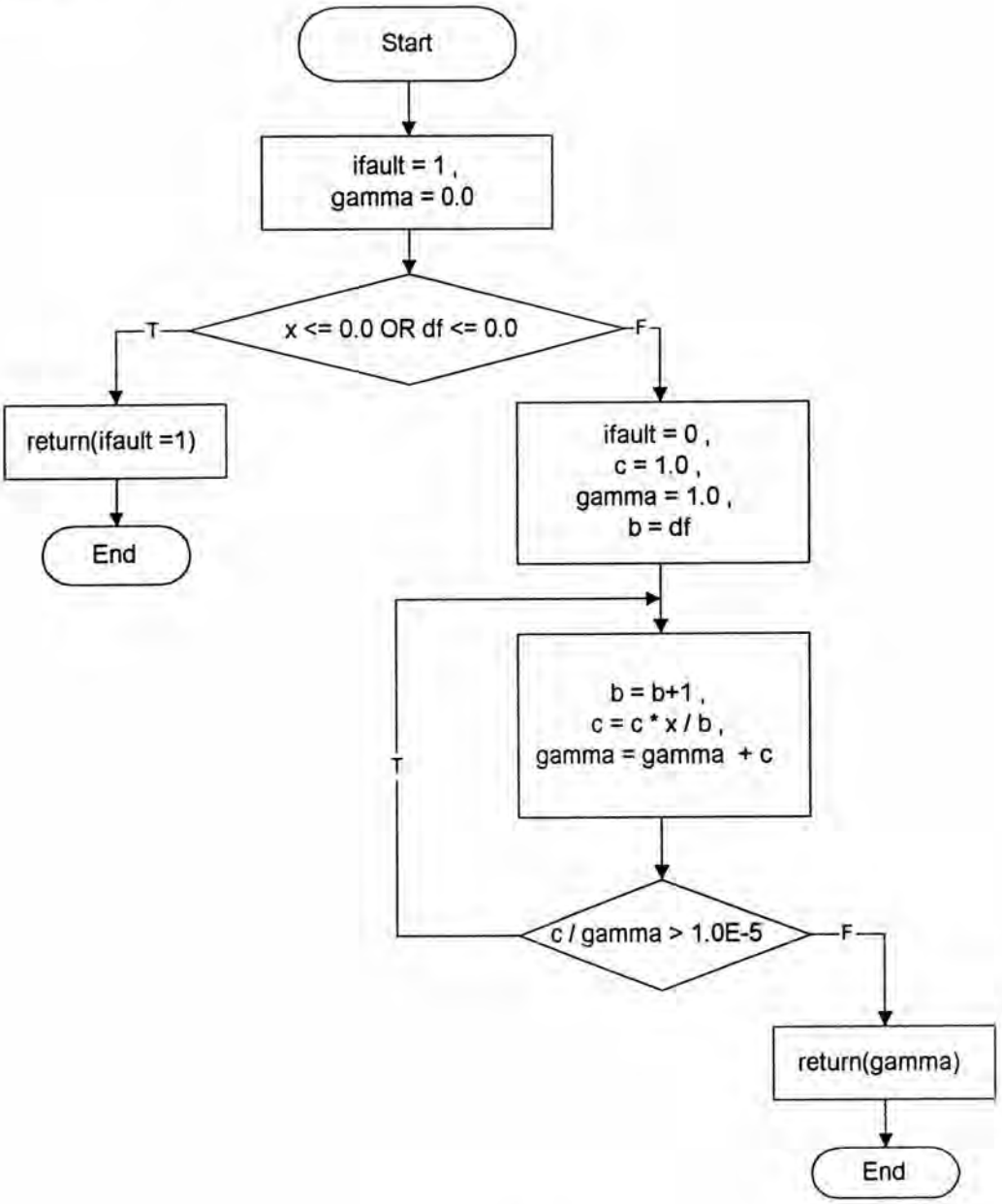


รูปภาพที่ 3.11

4. ฟังก์ชัน gamma

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงแกมมา ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย anyash และผังงานฟังก์ชัน gamma ซึ่งแสดงได้ดังรูปภาพที่ 3.12

<u>ฟังก์ชัน gamma</u>

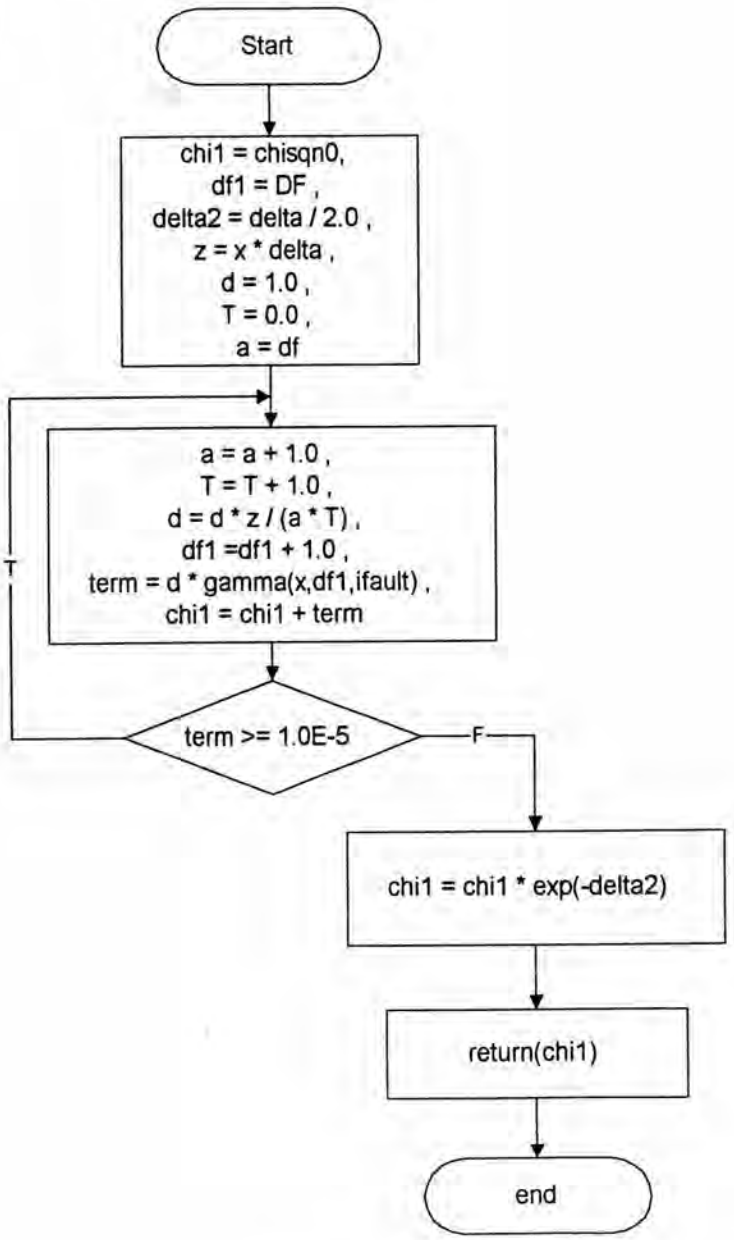


รูปภาพที่ 3.12

5. ฟังก์ชัน chi1

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงไคกำลังสอง ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย anyash และผังงานฟังก์ชัน chi1 ซึ่งแสดงได้ดังรูปภาพที่ 3.13

ฟังก์ชัน chi1

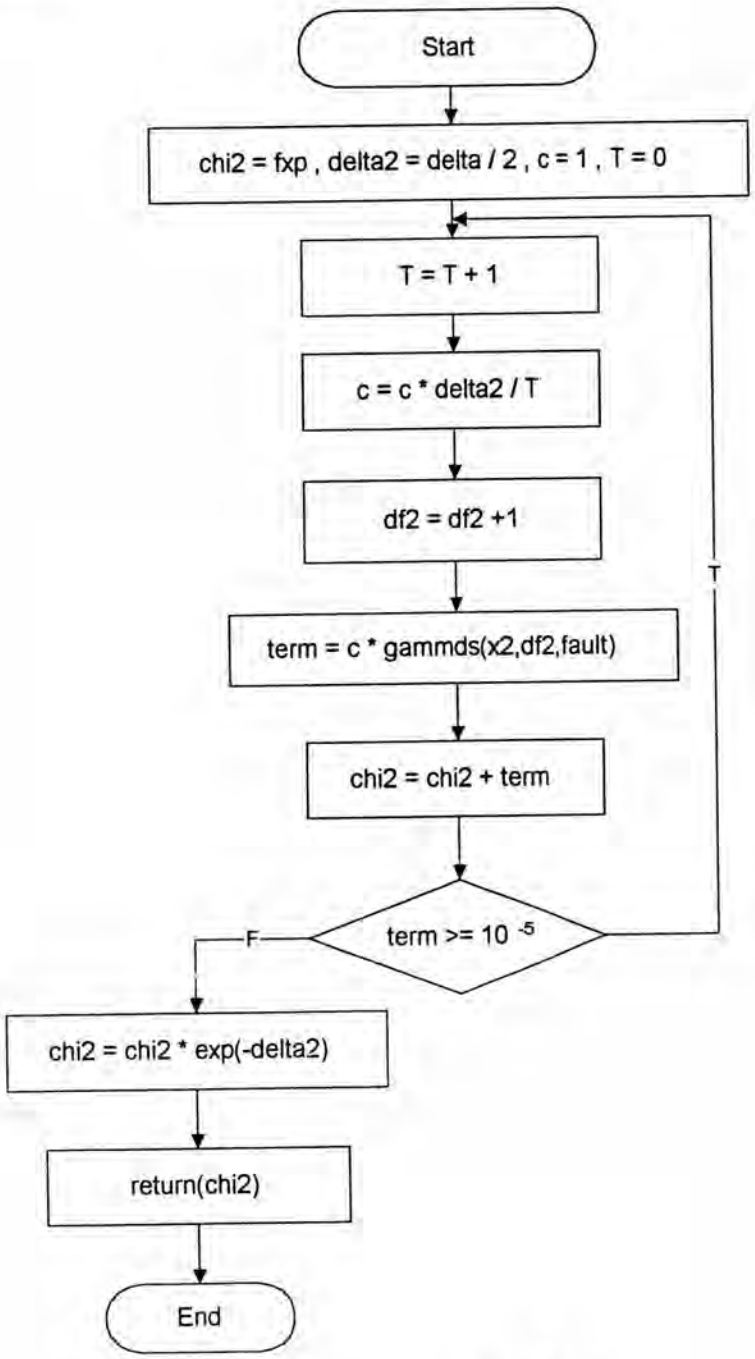


รูปภาพที่ 3.13

6. ฟังก์ชัน chi2

ฟังก์ชันนี้ทำหน้าที่คำนวณค่าฟังก์ชันการแจกแจงไคกำลังสอง ซึ่งโปรแกรมย่อยที่เรียกใช้งานฟังก์ชันนี้ได้แก่ โปรแกรมย่อย painaik และผังงานฟังก์ชัน chi2 ซึ่งแสดงได้ดังรูปภาพที่ 3.14

<u>ฟังก์ชัน chi2</u>



รูปภาพที่ 3.14