



โครงการ

เรื่อง ฟังก์ชันโปรแกรม คณิตศาสตร์

จัดทำโดย

นาย ธรรมรัตน์ เกิดลำเจียก

รหัสนิสิต 57160016 Sec 1

เสนอ

ผู้ช่วยศาสตราจารย์ นวลศรี เต็มวัฒนา

วิชาหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์

ภาคเรียนที่ 1 ปีการศึกษา 2557

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

อำเภอเมือง จังหวัดชลบุรี

ก. บทคัดย่อ

ชื่อเรื่อง	ฟังก์ชัน โปรแกรมคณิตศาสตร์
ชื่อผู้จัดทำ	นาย ธรรมรัตน์ เกิดลำเจียก
สาขาวิชา	วิศวกรรมซอฟต์แวร์
คณะ	วิทยาการสารสนเทศ
ปีการศึกษา	2557

โครงการนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งในรายวิชาหลักและวิธีการสำหรับวิศวกรรมซอฟต์แวร์ ของผู้ช่วยศาสตราจารย์นวลศรี เต็มวัฒนา ซึ่งเป็นการนำความรู้ที่ได้เรียนในห้องเรียนมาประยุกต์ใช้งานจริง ทำให้มีความเข้าใจในเนื้อหาของวิชานี้เพิ่มมากขึ้น

ซึ่งผู้จัดทำโครงการได้ออกแบบระบบของโปรแกรมให้ใช้ได้ง่ายและง่ายต่อการใช้งานสามารถนำไปใช้งานได้จริง ไม่ซับซ้อนมากจนเกินไป

กิตติกรรมประกาศ

โครงการเรื่องฟังก์ชันคณิตศาสตร์นี้สำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความอนุเคราะห์จากผู้ช่วยศาสตราจารย์ นวลศรี เด่นวัฒนา อาจารย์ผู้สอนในรายวิชา 888141 วิชาหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์ ที่ได้คอยให้คำปรึกษาและแนะนำวิธีการเขียนโปรแกรม

ขอขอบพระคุณอาจารย์ทุกท่านในสาขาวิศวกรรมซอฟต์แวร์ที่ได้ให้ความรู้จนโครงการเรื่องนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้จัดทำขอขอบพระคุณมา ณ โอกาสนี้

นาย ธรรมรัตน์ เกิดลำเจียก

ผู้จัดทำ

คำนำ

โครงการเล่มนี้เป็นส่วนหนึ่งของวิชา 888141 วิชาหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์

ชั้นปีที่ 1 คณะวิทยาการสารสนเทศ สาขาวิชาวิศวกรรมซอฟต์แวร์ มหาวิทยาลัยบูรพา โดยมีจุดประสงค์ เพื่อการศึกษาความรู้เกี่ยวกับ การเขียนโปรแกรมและการประยุกต์เขียนโปรแกรมให้สามารถใช้งานได้จริง และเกิดประโยชน์ ซึ่งโครงการเล่มนี้เป็นการศึกษาการเขียนโปรแกรมโดยการจำลองโปรแกรมคำนวณทางคณิตศาสตร์แบบง่าย ๆ

ผู้จัดทำได้เลือกหัวข้อนี้ในการทำโครงการ เนื่องจากเป็นเรื่องที่น่าสนใจ และเป็นสิ่งที่เราชอบเล่น ผู้จัดทำจึงได้ทำฟังก์ชัน และ โปรแกรมเกมส์คำนวณทางคณิตศาสตร์

นาย ธรรมรัตน์ เกิดลำเจียก

ผู้จัดทำ

สารบัญ

บทที่ 1 บทนำ	8
1.1 ที่มาของโครงการ	8
1.2 วัตถุประสงค์ของโครงการ	8
1.2.1 ด้านการศึกษา	8
1.2.2 ด้านการพัฒนา	8
1.3 ขอบเขตของโครงการ	8
1.4 เครื่องมือที่ใช้ในการพัฒนา	9
1.4.1 ความสามารถของซอฟต์แวร์	9
1.5 ขั้นตอนในการดำเนินงาน	9
1.6 ประโยชน์ที่คาดว่าจะได้รับ	10
1.6.1 ทรัพยากรทางด้านซอฟต์แวร์	10
1.6.2 ทรัพยากรทางด้านซอฟต์แวร์	10
บทที่ 2	11

ทฤษฎีและโครงการที่เกี่ยวข้อง	11
2.1 หลักและวิธีการโปรแกรมภาษาซี	11
2.1.1 โครงสร้างภาษาซี.....	11
2.1.2 ตัวแปรกับชนิดของข้อมูล	12
บทที่ 3	19
วิธีการดำเนินโครงการ.....	19
3.1 วิเคราะห์และออกแบบความสามารถของระบบ	19
3.1.1 ข้อกำหนดซอฟต์แวร์ในส่วนของผู้ใช้งาน	19
3.1.2 ออกแบบฐานข้อมูล	20
3.2 รหัสเทียมของโปรแกรม.....	21
3.3 แผนผังลำดับการทำงาน	31
บทที่ 4	35
4.1 โปรแกรมคำนวณทางคณิตศาสตร์	35
4.1.1 หน้าจอเริ่มต้นและเมนู	35
4.1.2 เลือกการคำนวณที่อยากให้กระทำเช่น ในเครื่องคิดเลข	
วิทยาศาสตร์	36
บทที่ 5	38

5.1	สรุปผลการดำเนินโครงการ.....	38
5.2	ประโยชน์ที่ได้รับ	38
5.3	ข้อจำกัด.....	38
	ภาคผนวก	39

สารบัญรูปภาพ

	หน้า
รูปที่ 1-1 แสดงการวางแผนการทำงาน	10
รูปที่ 2-1แสดงรูปแบบการรับข้อมูล	13
รูปที่ 2-2 ตัวดำเนินการทางคณิตศาสตร์	14
รูปที่ 4-1 หน้าจอเมื่อเข้าสู่โปรแกรม	35
รูปที่ 4-2 หน้าจอเมื่อเลือกเมนู 2 เพื่อเข้าใช้เครื่องคิดเลข	36
รูปที่ 4-3 หน้าจอเมื่อรับค่าไปคำนวณ	36
รูปที่ 4-4 หน้าจอเมื่อแสดงค่าฐานนิยม	37

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

เนื่องจากปัจจุบันทุกคนอยู่กับคณิตศาสตร์ มาอย่างช้านาน แล้วรายวิชานี้ได้สร้างความลำบากในการคำนวณ จึงมีเครื่องมือออกมาช่วยเหลือผู้ที่สนใจรายวิชานี้ ให้สามารถคำนวณหาผลลัพธ์ได้ง่ายขึ้น ดังนั้นข้าพเจ้า จึงมองเห็นโอกาสที่จะ ฝึกการใช้ภาษาซีลองจำลองทำฟังก์ชันคณิตศาสตร์ เพื่อคำนวณค่าแบบต่าง ๆ ออกมาให้ผู้ที่สนใจในคณิตศาสตร์

1.2 วัตถุประสงค์ของโครงการ

1.2.1 ด้านการศึกษา

➤ โครงการนี้เป็นส่วนหนึ่งของวิชาหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์ เป็นการนำเนื้อหาที่ได้เรียนรู้ นำมาเขียนโปรแกรมที่สามารถใช้งานได้ และเกิดประโยชน์

1.2.2 ด้านการพัฒนา

➤ เพื่อเป็นการทดลองและฝึกการเขียนโปรแกรม

➤ เพื่อเป็นการต่อยอดและพัฒนาซอฟต์แวร์ในอนาคต ต่อไป

1.3 ขอบเขตของโครงการ

โปรแกรมคำนวณทางคณิตศาสตร์ โปรแกรมที่จะนำเราไปสู่ทางเลือกของการคำนวณโดยแบ่งออกเป็น เครื่องคิดเลขทั่วไป เครื่องคิดเลขวิทยาศาสตร์ เครื่องคิดเลขทางสถิติ

ซึ่งโปรแกรมมีข้อจำกัดดังต่อไปนี้

➤ บางการคำนวณ จะสามารถทำได้แค่ประมาณค่าเท่านั้น เนื่องจากต้องการความรู้ทางคณิตศาสตร์ และ การโปรแกรมภาษาซี มากกว่านี้

- ถ้าเป็นการรับค่า จะทำได้ทีละขั้นตอนการคำนวณเท่านั้น

1.4 เครื่องมือที่ใช้ในการพัฒนา

เครื่องมือที่ใช้ในการพัฒนาการเขียนโปรแกรมคำนวณ และ ฟังก์ชันคณิตศาสตร์ มีดังต่อไปนี้

- การประกาศตัวแปร
- การรับค่าและแสดงผลที่หน้าจอ
- ข้อความสั่งประเภทเงื่อนไข
- ข้อความสั่งลูป
- ตัวแปรแถวลำดับ
- ฟังก์ชัน
- เพิ่มข้อมูลอักขระ
- เมคไฟล์
- สตริง

1.4.1 ความสามารถของซอฟต์แวร์

โปรแกรมคำนวณคณิตศาสตร์ ประกอบได้ด้วย 2 ส่วนหลักๆ ได้แก่

- คำนวณอย่างง่าย
- หาค่าตรีโกณ
- หาค่าทางสถิติ

1.5 ขั้นตอนในการดำเนินงาน

1. ศึกษาข้อมูล
2. วางแผน
3. วิเคราะห์

4. เขียนโปรแกรม
5. ทดสอบโปรแกรม
6. ทำรูปเล่ม
7. นำเสนอโครงงาน
8. ปรับแก้เอกสาร

รูปที่ 1-1 แสดงการวางแผนการทำงาน

กิจกรรม	27-31 ต.ค.				1-15 พ.ย.				16-30 พ.ย.				1-15 ธ.ค.				16-24 ธ.ค.			
1. ศึกษาข้อมูล			X	X																
2. วางแผน					X	X														
3. วิเคราะห์						X	X													
4.เขียนโปรแกรม							X	X	X	X										
5.ทดสอบโปรแกรม										X										
6.ทำรูปเล่ม											X	X								
7.นำเสนอโครงงาน												X								
8. ปรับแก้เอกสาร													X	X	X	X	X			

1.6 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำความรู้ที่ได้รับจากการเรียนวิชาหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์ มาประยุกต์ใช้ในการพัฒนาโปรแกรมคำนวณทางคณิตศาสตร์ ที่จำลองจากฟังก์ชันทางคณิตศาสตร์ในปัจจุบัน และเป็นการต่อยอดโครงงานเพื่อประโยชน์ต่อการศึกษาในอนาคต

1.6.1 ทรัพยากรทางด้านซอฟต์แวร์

- คอมพิวเตอร์พกพา จำนวน 1 เครื่อง

1.6.2 ทรัพยากรทางด้านซอฟต์แวร์

- โปรแกรม putty
- โปรแกรม winSCP

บทที่ 2

ทฤษฎีและโครงการที่เกี่ยวข้อง

โครงการฟังก์ชัน และ โปรแกรมทางคณิตศาสตร์ จัดทำขึ้นจากการนำความรู้จากการเรียนในรายวิชา วิชาหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์ (Principles of Programming for Software Engineering) และการเรียนวิชาปฏิบัติการหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์ (Principles of Programming for Software Engineering Laboratory) มาใช้ในการประกอบโครงการ

2.1 หลักและวิธีการโปรแกรมภาษาซี

ฟังก์ชัน และ โปรแกรมทางคณิตศาสตร์ เป็นโปรแกรมที่อาศัยความรู้เรื่องภาษาซี จากการเรียนในรายวิชาที่กล่าวมาข้างต้น

2.1.1 โครงสร้างภาษาซี

ภาษา C เป็นภาษาคอมพิวเตอร์ที่ถูกค้นคิดขึ้นโดย Denis Ritchie ในปี ค.ศ. 1970

ภาษา C เป็นภาษาคอมพิวเตอร์ชนิดคอมไพล์ (compiled Language) ซึ่งมีคอมไพเลอร์ (Compiler) ทำหน้าที่ในการคอมไพล์ (Compile) หรือแปลงคำสั่งทั้งหมดในโปรแกรมให้เป็นภาษาเครื่อง (Machine Language) เพื่อให้เครื่องคอมพิวเตอร์นำคำสั่งเหล่านั้นไปทำงานต่อไป

โครงสร้างของภาษา C

ทุกโปรแกรมของภาษา C มีโครงสร้างเป็นลักษณะดังนี้

เฮดเดอร์ไฟล์ (Header Files)

เป็นส่วนที่เก็บไลบรารีมาตรฐานของภาษา C ซึ่งจะถูกรวมเข้ามารวมกับโปรแกรมในขณะที่กำลังทำการคอมไพล์ โดยใช้คำสั่ง

#include<ชื่อเฮดเดอร์ไฟล์> หรือ #include "ชื่อเฮดเดอร์ไฟล์"

ส่วนตัวแปรแบบ Global (Global Variables)

เป็นส่วนที่ใช้ประกาศตัวแปรหรือค่าต่าง ๆ ที่ให้ใช้ได้ทั้งโปรแกรม ซึ่งใช้ได้ทั้งโปรแกรม ซึ่งในส่วนไม่จำเป็นต้องมีก็ได้

ฟังก์ชัน (Functions)

เป็นส่วนที่เก็บคำสั่งต่างๆ ไว้ ซึ่งในภาษา C จะบังคับให้มีฟังก์ชันอย่างน้อย 1 ฟังก์ชันนั่นคือ ฟังก์ชัน Main และในโปรแกรม 1 โปรแกรมสามารถมีฟังก์ชันได้มากกว่า 1 ฟังก์ชัน

ส่วนตัวแปรแบบ Local (Local Variables)

เป็นส่วนที่ใช้สำหรับประกาศตัวแปรที่จะใช้ในเฉพาะฟังก์ชันของตนเอง ฟังก์ชันอื่นไม่สามารถเข้าถึงหรือใช้ได้ ซึ่งจะต้องทำการประกาศตัวแปรก่อนการใช้งานเสมอ และจะต้องประกาศไว้ในส่วนนี้เท่านั้น

ตัวแปรโปรแกรม (Statements)

เป็นส่วนที่อยู่ถัดลงมาจากส่วนตัวแปรภายใน ซึ่งประกอบไปด้วยคำสั่งต่าง ๆ ของภาษา C และคำสั่งต่าง ๆ จะใช้เครื่องหมาย ; เพื่อเป็นการบอกให้รู้ว่าจบคำสั่งหนึ่ง ๆ แล้ว ภาษา C จะแยกความแตกต่างของตัวพิมพ์เล็กและพิมพ์ใหญ่หรือ Case Sensitive นั่นเอง

ค่าส่งกลับ (Return Value)

เป็นส่วนที่บอกให้รู้ว่า ฟังก์ชันนี้จะส่งค่าอะไรกลับไปให้กับฟังก์ชันที่เรียกฟังก์ชัน ซึ่งเรื่องนี้จะยกไปกล่าวในเรื่องฟังก์ชันอย่างละเอียดอีกทีหนึ่ง

หมายเหตุ (Comment)

เป็นส่วนที่ใช้สำหรับแสดงข้อความเพื่ออธิบายสิ่งที่ต้องการในโปรแกรม ซึ่งจะใช้เครื่องหมาย /* และ */ ปิดหัวและปิดท้ายของข้อความที่ต้องการ

2.1.2 ตัวแปรกับชนิดของข้อมูล

ตัวแปร คือ ชื่อที่กำหนดขึ้นเพื่อใช้ในการเก็บข้อมูล ซึ่งการประกาศตัวแปรขึ้นมาใช้งานจะเป็นไปตามกฎการตั้งชื่อ

ชนิดของข้อมูล คือ สิ่งที่ใช้กำหนดลักษณะและขอบเขตของข้อมูลนั้นๆ โดยข้อมูลที่มีชนิดของข้อมูลแตกต่างกัน ก็จะเก็บข้อมูลได้ในลักษณะแตกต่างกัน และขอบเขตของข้อมูลที่เก็บได้ก็จะไม่เท่ากัน ซึ่งในภาษาซีแบ่งชนิดของข้อมูลออกเป็น 4 แบบ คือ

ชนิดข้อมูลแบบไม่มีค่า หรือ Void Type (Void)

ข้อมูลชนิดนี้ จะไม่มีค่าและจะไม่ใช้ในการกำหนดชนิดตัวแปร แต่ส่วนใหญ่จะใช้เกี่ยวกับฟังก์ชัน ซึ่งจะขอยกไปอธิบายในเรื่องฟังก์ชัน

ชนิดข้อมูลแบบจำนวนเต็ม หรือ Integer Type (int)

เป็นชนิดข้อมูลที่เป็นตัวเลขจำนวนเต็ม ไม่มีทศนิยม ซึ่งภาษา C จะแบ่งข้อมูลชนิดนี้ออกได้เป็น 3 ระดับ คือ short int, int และ long int

ชนิดข้อมูลแบบอักขร หรือ Character Type (char)

ข้อมูลชนิดนี้ก็คือ ตัวอักษรตั้งแต่ A-Z เลข 0-9 และสัญลักษณ์ต่าง ๆ ตามมาตรฐาน ASCII (American Standard Code Information Interchange) ซึ่งเมื่อกำหนดให้กับตัวแปรแล้วตัวแปรนั้นจะรับค่าได้เพียง 1 ตัวอักษรเท่านั้น และสามารถรับข้อมูลจำนวนเต็มตั้งแต่ถึง 127 จะใช้ขนาดหน่วยความจำ 1 ไบต์

ชนิดข้อมูลแบบอักขร หรือ Character Type (char)

ข้อมูลชนิดนี้ก็คือ ตัวอักษรตั้งแต่ A-Z เลข 0-9 และสัญลักษณ์ต่าง ๆ ตามมาตรฐาน ASCII (American Standard Code Information Interchange) ซึ่งเมื่อกำหนดให้กับตัวแปรแล้วตัวแปรนั้นจะรับค่าได้เพียง 1 ตัวอักษรเท่านั้น และสามารถรับข้อมูลจำนวนเต็มตั้งแต่ถึง 127 จะใช้ขนาดหน่วยความจำ 1 ไบต์หรือ 8 บิต

ชนิดข้อมูลแบบทศนิยม หรือ Floating Point Type (float)

เป็นข้อมูลชนิดตัวเลขที่มีจุดทศนิยม ซึ่งสามารถแบ่งออกเป็น 3 ระดับ คือ float, double และ long double

2.1.3 ข้อความสั่งการรับค่า และการแสดงผลข้อมูล

ในภาษา C จะมีฟังก์ชันสำหรับการพิมพ์และการรับข้อมูลไว้ให้ใช้มากมาย การพิมพ์ก็คือ การนำข้อมูลไปแสดงที่หน้าจอ และการรับข้อมูลก็คือ การรับข้อมูลจากคีย์บอร์ดเข้ามา ซึ่งในภาษา C นั้นจะมองอุปกรณ์ในเครื่องคอมพิวเตอร์เป็นไฟล์ทั้งหมด

2.1.3.1 การแสดงผล

ฟังก์ชันมาตรฐานที่ใช้ในการพิมพ์ข้อมูลของภาษา C คือ printf

2.1.3.2 การรับค่า

ฟังก์ชันมาตรฐานที่ใช้ในการรับข้อมูลของภาษา C คือ scanf

รูปที่ 2-1 แสดงรูปแบบการรับข้อมูล

รหัสรูปแบบ	ชนิดข้อมูล
%c	ตัวอักษรหนึ่งตัว(char)
%d	จำนวนเต็ม (int)
%ld	จำนวนเต็ม (long int)
%f	จำนวนทศนิยม(float)

2.1.4 ตัวดำเนินการทางคณิตศาสตร์

เครื่องหมายคณิตศาสตร์ และในทางคอมพิวเตอร์แต่ละตัวจะมีค่าความสำคัญแตกต่างกัน ซึ่งเวลาใช้จะต้องคำนึงถึงค่าสำคัญของเครื่องหมายแต่ละตัวด้วย หากต้องการให้ตัวดำเนินการที่มีความสำคัญน้อยกว่าประมวลผลก่อน จะต้องใส่เครื่องหมายวงเล็บ () ด้วย

รูปที่ 2-2 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ความหมาย
+	การบวก(Addition)
-	การลบ(Subtraction)
*	การคูณ(Multiply)
/	การหาร(Divide)
%	การหารเอาเศษ(Modulus)

2.1.5 การโปรแกรมแบบมีเงื่อนไข

โปรแกรมแบบมีเงื่อนไข คือการเปรียบเทียบค่าเท็จจริง จากการประมวลผล

2.1.5.1 คำสั่ง 2 ทางเลือก

คำสั่ง 2 ทางเลือกเป็นพื้นฐานของคำสั่งเงื่อนไขในภาษาคอมพิวเตอร์นั้น คำสั่งประเภทนี้จะต้องมีเงื่อนไขการตัดสินใจ เพื่อใช้หาคำตอบว่าจะไปทางไหน ถ้าเงื่อนไขเป็นจริงก็จะไปหาคำสั่งทางหนึ่ง แต่ถ้าเงื่อนไขเป็นเท็จก็จะไปหาคำสั่งอีกทางหนึ่ง ผังการทำงานของคำสั่ง 2 ทางเลือก

If...else คำสั่ง if...else นี้ จะต้องใช้เงื่อนไขเพื่อใช้เลือกจะทำคำสั่งไหน ถ้าเงื่อนไขเป็นจริงก็จะไปทำคำสั่งที่ 1 แต่เงื่อนไขเป็นเท็จก็จะไปทำคำสั่งที่ 2 ซึ่งไม่มีโอกาสที่จะทำทั้งสองคำสั่งได้เลย

2.1.5.2 คำสั่งหลายทางเลือก

นอกจากคำสั่ง 2 ทางเลือกแล้ว ภาษา C ยังมีคำสั่งหลายทางเลือกให้ใช้ด้วยเพื่อความสะดวก รวดเร็วในการเขียนและการทำความเข้าใจ เพราะไม่ต้องมานั่งเขียนโปรแกรมที่ยาว

Switch เป็นคำสั่งที่แปลงมาจากคำสั่ง Nested if คำสั่งนี้จะมีตัวแปรหนึ่งตัวที่ใช้หาว่าจะไปทำ ที่คำสั่งไหนหรือ case ไหน ซึ่งในรูปที่ 2-3 แสดงผังการทำงานคำสั่ง switch ผู้ใช้สามารถสร้าง case ให้มีจำนวนตามต้องการได้

2.1.6 การโปรแกรมแบบทำซ้ำ

2.1.6.1 การทำซ้ำทดสอบเงื่อนไขก่อน (Pre-test loop) และทดสอบเงื่อนไขทีหลัง (Post-test loop)
ในหลักการโปรแกรมมีการทดสอบหรือประเมินเงื่อนไขสองแบบ คือ

- แบบทดสอบเงื่อนไขก่อน คือ การทำสอบเงื่อนไขก่อนจะมีการประมวลผล
- แบบทดสอบเงื่อนไขทีหลัง คือ การทำสอบเงื่อนไขหลังจากที่มีการประมวลผลแล้ว

2.1.6.2 คำสั่ง for

เป็นคำสั่งวนซ้ำแบบ pre-test loop คือ ประมวลผลก่อนการทำซ้ำเหมาะกับการทำซ้ำแบบรู้จำนวนรอบในการทำซ้ำ

2.1.6.3 คำสั่ง while

เป็นคำสั่งทำซ้ำแบบ pre-test loop คือ จะมีการตรวจสอบเงื่อนไขก่อนที่จะทำซ้ำ ถ้าเงื่อนไขเป็นจริง จึงจะเกิดการซ้ำ

2.1.6.4 คำสั่ง do..while

เป็นคำสั่งแบบ post-test loop คือ จะมีการทำตามคำสั่งก่อน แล้วจึงตรวจสอบเงื่อนไข โดยถ้าเงื่อนไขเป็นจริง จึงจะมีการทำซ้ำ

2.1.7 ฟังก์ชัน

2.1.7.1 การเขียนโปรแกรมแบบฟังก์ชัน

การเขียนโปรแกรมแบบฟังก์ชันมีขั้นตอนดังนี้

1. ประกาศฟังก์ชันที่ต้องการไว้ในส่วนแรกของโปรแกรม
2. เรียกใช้ฟังก์ชันที่ประกาศไว้
3. กำหนดหรือสร้างรายละเอียดของฟังก์ชันนั้น

2.1.7.2 รูปแบบของฟังก์ชัน

รูปแบบของฟังก์ชันของภาษา ซี มีอยู่ 4 รูปแบบ

แบบที่ 1 ฟังก์ชันแบบไม่มีการส่งค่ากลับ และไม่มีการพารามิเตอร์ เป็นฟังก์ชันที่ไม่มีการส่งค่ากลับไปที่ฟังก์ชันที่เรียกมา และไม่มีการส่งค่าจากฟังก์ชันที่เรียกมาไปด้วย

แบบที่ 2 ฟังก์ชันแบบไม่มีการส่งค่ากลับ และพารามิเตอร์ เป็นฟังก์ชันที่จะไม่มีการส่งค่ากลับไปที่ฟังก์ชันที่เรียกขึ้นมา แต่มีการส่งค่าจากฟังก์ชันที่เรียกมาไปด้วย

แบบที่ 3 ฟังก์ชันแบบมีการส่งค่ากลับ และไม่มีการพารามิเตอร์ เป็นฟังก์ชันที่จะมีการส่งค่ากลับไปที่ฟังก์ชันที่เรียกมา แต่ไม่มีการส่งค่าจากฟังก์ชันที่เรียกมาไปด้วย

แบบที่ 4 ฟังก์ชันแบบมีการส่งค่ากลับ และมีพารามิเตอร์ เป็นฟังก์ชันที่จะมีการส่งค่ากลับไปที่ฟังก์ชันที่เรียกมา แต่มีการส่งค่าจากฟังก์ชันที่เรียกมาไปด้วย

2.1.8 ตัวแปรแถวลำดับ

ตัวแปรแถวลำดับ (Array) คือ ตัวแปรที่สามารถเก็บข้อมูลได้เป็นชุด โดยสร้างตัวแปรขึ้นมาเพียงตัวเดียว แต่ข้อมูลนั้นต้องเป็นชนิดเดียวกัน

2.1.9 การโปรแกรมกับแฟ้มข้อมูลอักขระ

แฟ้มข้อมูล (File) คือ ที่เก็บข้อมูลถาวรก่อนหน้านี้ ข้อมูลที่เรารับค่าและแสดงผลจะถูกบันทึกอยู่ในที่เก็บข้อมูลชั่วคราวเท่านั้น เมื่อโปรแกรมทำงานจบ ข้อมูลเหล่านั้นจะหายไป แต่ไฟล์ จะทำให้เราสามารถเก็บข้อมูลสำหรับงานของเราไว้ได้อย่างถาวร เมื่อใดที่เราต้องการเข้าถึงข้อมูลที่ถูกบันทึกไว้ เราจึงสั่งให้โปรแกรมเข้าไปนำค่านั้นๆ ออกมาใช้งาน

2.1.9.1 การเปิดแฟ้ม

- ต้องประกาศพอยน์เตอร์สำหรับชี้ไปยังแฟ้มที่เราต้องการอ่าน
- ระบุชื่อแฟ้ม
- บอกจุดประสงค์ของการเปิดแฟ้ม

โดยที่

r เปิดแฟ้มข้อมูลเพื่ออ่าน

W เปิดแฟ้มข้อมูลเพื่อเขียนทับไฟล์เก่า

a เปิดแฟ้มข้อมูลเพื่อเขียนต่อ

FILE *fp

```
fp = fopen(“ชื่อแฟ้ม”, “จุดประสงค์การเปิด”);
```

2.1.9.2 การปิดแฟ้ม

เมื่อทำงานกับแฟ้มข้อมูลใดๆ เสร็จเรียบร้อยแล้ว เราควรทำการปิดแฟ้มข้อมูล (close) เพื่อทำการบันทึกข้อมูลสุดท้ายลงในแฟ้มข้อมูลและทำการคืนทรัพยากรให้กับเครื่อง

```
fclose(fp)
```

2.1.10 การกำหนดชนิดโครงสร้าง

โครงสร้างหมายถึงองค์ประกอบของข้อมูลที่เกี่ยวข้องกัน ซึ่งแต่ละองค์ประกอบอาจมีชนิดของข้อมูลที่เหมือนหรือแตกต่างกันก็ได้ ในเรื่องแถวลำดับจะเห็นได้ว่าสมาชิกทุกตัวต้องเป็นชนิดเดียวกัน แต่ถ้าโครงสร้างสมาชิกหรือองค์ประกอบต่างๆอาจเป็นชนิดที่แตกต่างกันได้

- ฟิลด์ คือ ชนิดของตัวแปรต่างๆจะเป็น int, char, float ฯลฯ
- ตัวแปรลิสต์ คือ ชื่อตัวแปรที่อยู่ภายใต้วงเล็บปีกกาของโครงสร้าง {}

```
typedef struct ชื่อตัวแปรโครงสร้าง
```

```
{
```

```
    ชื่อฟิลด์ 1 ชื่อตัวแปรลิสต์ 1;
```

```
    ..
```

```
    ชื่อฟิลด์ n ชื่อตัวแปรลิสต์ n;
```

```
};
```

การแสดงค่าทางจอภาพตัวแปรแบบตัวชี้

```
printf(“%ชนิดของข้อมูล”, ชื่อตัวแปรโครงสร้าง.ชื่อตัวแปรลิสต์);
```

2.1.11 สตริง

สตริง หมายถึง เซตของอักขระที่เรียงต่อกัน ซึ่งอาจมีความยาวที่แน่นอนหรือความยาวที่เปลี่ยนแปลงได้ กล่าวคือ char หลายๆตัวเรียงต่อกัน ซึ่งในสตริงในภาษาซีจะถูกเก็บในตัวแปรประเภทแถวลำดับ ซึ่งมีความยาวแปรเปลี่ยนได้และใช้อักขระ Null แทนจุดสิ้นสุดสตริง

การประกาศตัวแปรแบบสตริงประกาศได้ 2 แบบ คือ

1. ประกาศเป็นแถวลำดับอักขระ
2. ประกาศเป็นชนิดตัวชี้

บทที่ 3

วิธีการดำเนินโครงการ

การดำเนินการออกแบบและเขียนฟังก์ชัน และ โปรแกรมคำนวณทางคณิตศาสตร์ มีวิธีการดำเนินโครงการต่อไปนี้

3.1 วิเคราะห์และออกแบบความสามารถของระบบ

ลองวิเคราะห์ก่อนว่าโปรแกรมในการคำนวณจะต้องมีการทำงานในส่วนใดบ้าง และมีส่วนไหนที่จะต้องเชื่อมโยงกัน โดยผู้จัดทำได้เล็งเห็นว่าควรแบ่งออกเป็น 3 รูปแบบเพื่อง่ายต่อการจำแนก และการคำนวณค่า

3.1.1 ข้อกำหนดซอฟต์แวร์ในส่วนของผู้ใช้งาน

ในส่วนของผู้ใช้งานเข้ามาถึงแล้วมีการเลือกเมนูว่าต้องการทำอะไร ความสามารถในการใช้งานโปรแกรมของผู้ใช้ แบ่งเป็นข้อๆ ได้ดังนี้

1. เมนูหลัก
 - 1.1. เครื่องคิดเลขทั่วไป
 - 1.2. เครื่องคิดเลขวิทยาศาสตร์
 - 1.3. เครื่องคิดเลขทางสถิติ
 - 1.4. ออกจากโปรแกรม
2. เครื่องคิดเลขทั่วไป
 - 2.1. บวก
 - 2.2. ลบ
 - 2.3. คูณ
 - 2.4. หาร
3. เครื่องคิดเลขวิทยาศาสตร์
 - 3.1. ยกกำลัง
 - 3.2. แพรกโทเรียล
 - 3.3. พิโบนาซชี
 - 3.4. ไซน์

- 3.5. โคไซน์
- 3.6. แทนเจนต์
- 3.7. โคเซค
- 3.8. เซค
- 3.9. คอท

4. เครื่องคิดเลขทางสถิติ

- 4.1. ค่าสูงสุด
- 4.2. ค่าต่ำสุด
- 4.3. พิสัย
- 4.4. ค่าเฉลี่ย
- 4.5. ค่ากึ่งกลาง
- 4.6. ค่าฐานนิยม
- 4.7. เรียงค่ารับเข้า

3.1.2 ออกแบบฐานข้อมูล

- 1. ใช้ภาษา C ในการเขียนโปรแกรม
- 2. เรียกใช้ไลบรารี <stdio.h> และ <string.h>
- 3. แยกฟังก์ชันในการเขียนโปรแกรม
- 4. มีการรับข้อมูลจากผู้ใช้

3.2 รหัสเทียมของโปรแกรม

❖ Main function

Algorithm function: Main

➤ Input: รับค่า หนึ่ง ถึง สองค่าทางแป้นพิมพ์เพื่อดำเนินการทางคณิตศาสตร์

➤ Output: แสดงผลลัพธ์ออกทางหน้าจอ

1. ประกาศตัวแปร ชื่อ a,b,c,r1,r2 ประเภท float
2. ประกาศตัวแปร ชื่อ sym ประเภท char
3. ประกาศตัวแปร ชื่อ menu,rmenu,smenu ประเภท int
4. เข้าสู่ทางเลือกเมนู Menu

4.1 Regular Calculator

4.1.1 เลือกเมนู Menu

4.1.1.1 Plus บวก

- 4.1.1.1.1 เรียกใช้ฟังก์ชัน input รับค่า a และ b
- 4.1.1.1.2 เรียกใช้ฟังก์ชัน plus โดยส่งค่า a และ b
- 4.1.1.1.3 เรียกใช้ฟังก์ชัน print โดย ส่งค่า a,b และ c

4.1.1.2 Minus ลบ

- 4.1.1.2.1 เรียกใช้ฟังก์ชัน input รับค่า a และ b
- 4.1.1.2.2 เรียกใช้ฟังก์ชัน minus โดยส่งค่า a และ b
- 4.1.1.2.3 เรียกใช้ฟังก์ชัน print โดย ส่งค่า a,b และ c

4.1.1.3 Multiply คูณ

- 4.1.1.3.1 เรียกใช้ฟังก์ชัน input รับค่า a และ b
- 4.1.1.3.2 เรียกใช้ฟังก์ชัน mult โดยส่งค่า a และ b
- 4.1.1.3.3 เรียกใช้ฟังก์ชัน print โดย ส่งค่า a,b และ c

4.1.1.4 Divind ทหาร

- 4.1.1.4.1 เรียกใช้ฟังก์ชัน input รับค่า a และ b
- 4.1.1.4.2 เรียกใช้ฟังก์ชัน divind โดยส่งค่า a และ b

4.1.1.4.3 เรียกใช้ฟังก์ชัน print โดย ส่งค่า a,b และ c

4.1.1.5 ออกสู่เมนูหลัก Main Menu

4.2 Scientific Calculator

4.2.1 เลือเมนู Menu

4.2.1.1 Power function

4.2.1.1.1 เรียกใช้ฟังก์ชัน input รับค่า a และ b

4.2.1.1.2 เรียกใช้ฟังก์ชัน power โดยส่งค่า a และ b

4.2.1.1.3 เรียกใช้ฟังก์ชัน print โดย ส่งค่า a,b และ c

4.2.1.2 Factorial Series

4.2.1.2.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.2.2 เรียกใช้ฟังก์ชัน factorial โดยส่งค่า a

4.2.1.2.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.3 Fibonacci Series

4.2.1.3.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.3.2 เรียกใช้ฟังก์ชัน fibonacci โดยส่งค่า a

4.2.1.3.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.4 Sine

4.2.1.4.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.4.2 เรียกใช้ฟังก์ชัน sine โดยส่งค่า a

4.2.1.4.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.5 Cosine

4.2.1.5.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.5.2 เรียกใช้ฟังก์ชัน cosine โดยส่งค่า a

4.2.1.5.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.6 Tangent

4.2.1.6.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.6.2 เรียกใช้ฟังก์ชัน sine โดยส่งค่า a ส่วน ฟังก์ชัน cosine โดยส่งค่า a

4.2.1.6.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.7 Cosec

4.2.1.7.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.7.2 เรียกใช้ฟังก์ชัน 1 ส่วน sine โดยส่งค่า a

4.2.1.7.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.8 Sec

4.2.1.8.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.8.2 เรียกใช้ฟังก์ชัน 1 ส่วน cosine โดยส่งค่า a

4.2.1.8.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.9 Cot

4.2.1.9.1 เรียกใช้ฟังก์ชัน input รับค่า a

4.2.1.9.2 เรียกใช้ฟังก์ชัน 1 ส่วน tangent โดยส่งค่า a

4.2.1.9.3 เรียกใช้ฟังก์ชัน sprintf โดย ส่งค่า a และ c

4.2.1.10 ออกสู่เมนูหลัก Main Menu

4.3 Accountant Calculator

4.3.1 ประกาศตัวแปร n,l,j,amenu,temp ประเภท int

4.3.2 ประกาศตัวแปร rest,max,min,x_bar,sum,med,mod,count ประเภท float

4.3.3 รับค่า n ผ่านฟังก์ชัน input เพื่อกำหนดค่า term

4.3.3.1 ประกาศตัวแปร set[n] ประเภท float และ numtemp[n] ประเภท int

4.3.3.1.1 รับค่าผ่านฟังก์ชัน inputAry ตามจำนวนค่า n

- 4.3.3.1.2 เรียงค่าจากมากไปน้อย Number-Sort
- 4.3.3.1.3 กำหนด Max = set[0]
- 4.3.3.1.4 กำหนด Min = set[n-1]
- 4.3.3.1.5 หา \bar{x} โดยนำค่า set ทั้งหมดบวกกันหารด้วย n
- 4.3.3.1.6 หาค่า med โดย นำ n+1 หาร 2 แล้วนับตำแหน่ง Array มาตอบ
- 4.3.3.1.7 หาค่า mode โดยการนับค่าที่มากที่สุดเป็นฐานนิยม ถ้ามีมากกว่าหนึ่งค่า เท่ากับไม่มีฐานนิยม
- 4.3.3.1.8 แสดงเมนู Menu
 - 4.3.3.1.8.1 แสดงค่า Max
 - 4.3.3.1.8.1.1 เรียกใช้ฟังก์ชัน Aryprint ส่งค่า max
 - 4.3.3.1.8.2 แสดงค่า Min
 - 4.3.3.1.8.2.1 เรียกใช้ฟังก์ชัน Aryprint ส่งค่า min
 - 4.3.3.1.8.3 แสดงค่าเฉลี่ย \bar{X}
 - 4.3.3.1.8.3.1 เรียกใช้ฟังก์ชัน Aryprint ส่งค่า \bar{x}
 - 4.3.3.1.8.4 แสดงค่าพิสัย Range
 - 4.3.3.1.8.4.1 เรียกใช้ฟังก์ชัน Aryprint ส่งค่า max ลบ min
 - 4.3.3.1.8.5 แสดงค่ากึ่งกลาง Med
 - 4.3.3.1.8.5.1 เรียกใช้ฟังก์ชัน Aryprint ส่งค่า med
 - 4.3.3.1.8.6 แสดงค่าฐานนิยม Mode
 - 4.3.3.1.8.6.1 เรียกใช้ฟังก์ชัน Aryprint ส่งค่า mode
 - 4.3.3.1.8.7 แสดงตัวเลขเรียงแล้ว (Sort)
 - 4.3.3.1.8.7.1 แสดงค่า set[n] ทั้งหมด
 - 4.3.3.1.8.8 เซ็ตค่าตัวเลขใหม่
 - 4.3.3.1.8.8.1 ขึ้นไปรับค่าทั้งหมด

4.3.3.1.8.9 กลับสู่เมนูหลัก Main menu

4.4 Read Help and Notice

4.4.1 ประกาศตัวแปร text ประเภท char

4.4.2 เปิดไฟล์ User_helping เพื่ออ่าน

4.4.3 แสดงข้อความในไฟล์ทางจอภาพ

4.4.4 ปิดไฟล์

4.4.5 กลับสู่เมนูหลัก Main Menu

4.5 ออกจากโปรแกรม Exit

5. จบการทำงาน End

❖ Sub-function

Algorithm function: Input,InputAry

- Input: รับค่าหนึ่งค่าทางแป้นพิมพ์
- Return: ตัวแปร key
 1. ค่ารับเข้าประเภท สตริง(string) ชื่อ inname
 2. ประกาศตัวแปร ชื่อ key ประเภท float และ ชื่อ temp ประเภท string
 3. แสดงข้อความจาก inname ทางหน้าจอ
 4. รับค่าทางคีย์บอร์ด
 5. เช็คข้อมูลรับเข้า ถ้าผิดประเภท ขึ้นเตือนและให้รับค่าใหม่
 6. ถูกประเภทให้ ส่งค่า (return) key กลับ

Algorithm function: Plus,Minus,Mult,Divind

- Input: รับค่าสองค่าทางพารามิเตอร์
- Return: ตัวแปร result
 - Function plus
 1. ค่ารับเข้า adder,addin ประเภท float
 2. ประกาศตัวแปร result ประเภท float
 3. result เท่ากับ adder บวก addin
 4. คืนค่า (return) result
 - Function minus
 1. ค่ารับเข้า miner,minin ประเภท float
 2. ประกาศตัวแปร result ประเภท float
 3. result เท่ากับ miner ลบ minin
 4. คืนค่า (return) result

- Function mult
 1. ค่ารับเข้า multer,multin ประเภท float
 2. ประกาศตัวแปร result ประเภท float
 3. result เท่ากับ multer คูณ multin
 4. คืนค่า (return) result
- Function divid
 1. ค่ารับเข้า divider,dividin ประเภท float
 2. ประกาศตัวแปร result ประเภท float
 3. result เท่ากับ divider ทหาร dividin
 4. คืนค่า (return) result

Algorithm function: factorial

- Input: รับค่าหนึ่งค่าทางพารามิเตอร์
- Return: ตัวแปร term*factorial(term-1)
 1. ค่ารับเข้า term ประเภท int
 2. If term = 0: ส่งค่า 1 กลับ
 3. Else: คืนค่า เป็น term คูณ factorial(term-1)

Algorithm function: fib

- Input: รับค่าหนึ่งค่าทางพารามิเตอร์
- Return: ตัวแปร fib(term-1)+fib(term-2)
 1. ค่ารับเข้า term ประเภท int
 2. If term = 1: ส่งค่า 0 กลับ

3. If term = 2: ส่งค่า 1 กลับ
4. Else: ส่งค่า fib(term-1) บวก fib(term-2)

Algorithm function: power

- Input: รับค่าสองค่าทางพารามิเตอร์
- Return: ตัวแปร mem
 1. ค่ารับเข้าชื่อ base ประเภท float และชื่อ pow ประเภท int
 2. ประกาศตัวแปร i ประเภท int และ mem ประเภท float
 3. กำหนด mem และ i เท่ากับ 0
 4. กำหนดลูป For(i=0;i<pow;i++)
 5. ภายใน ลูป ให้ mem เท่ากับ mem คูณ base
 6. คืนค่า mem

Algorithm function: sine

- Input: รับค่าหนึ่งค่าทางพารามิเตอร์
- Return: ตัวแปร sin
 1. ค่ารับเข้า radius ประเภท float
 2. ประกาศตัวแปร val,sin ประเภท float
 3. Val เท่ากับ radius คูณ (PI/180)
 4. Sin เท่ากับ val ลบ ฟังก์ชัน power ส่วน factorial ตามอนุกรม
 5. คืนค่า sin

Algorithm function: cosine

- Input: รับค่าหนึ่งค่าทางพารามิเตอร์
- Return: ตัวแปร cos
 1. ค่ารับเข้า radius ประเภท float
 2. ประกาศตัวแปร val, cosine ประเภท float
 3. Val เท่ากับ radius คูณ (PI/180)
 4. Cos เท่ากับ val ลบ ฟังก์ชัน power ส่วน factorial ตามอนุกรม
 5. คืนค่า cos

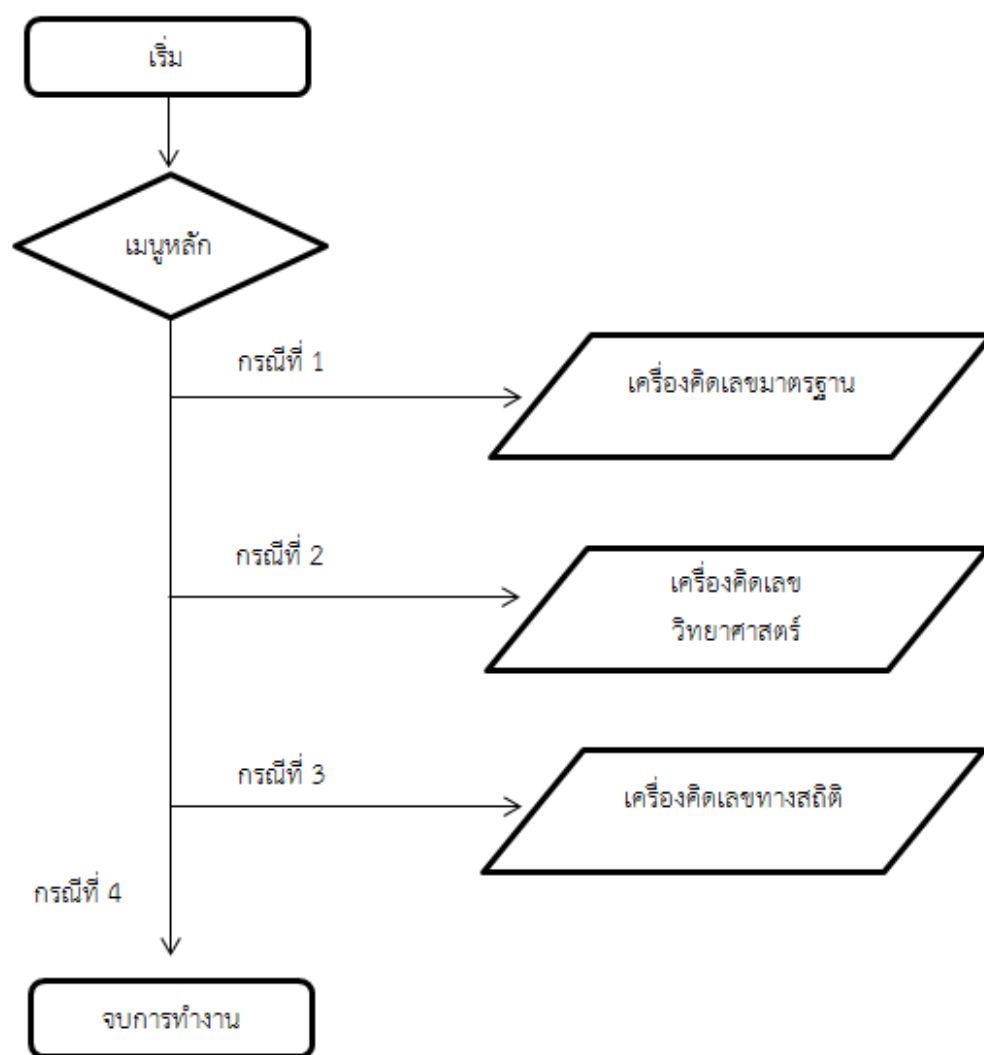
Algorithm function: sprintf, sprintf, print, Aryprint

- ✓ คือ ฟังก์ชัน แสดงค่าทางจอภาพตาม output ที่ออกแบบไว้
- Input: รับค่าหลายค่าทางคีย์บอร์ด
- Return: ไม่มีค่ารีเทิร์น
- Function sprintf
 1. รับค่า screen, sym ประเภท string และ ini, res ประเภท int
 2. แสดงค่าทั้งหมดทางจอภาพ ตาม output ที่ออกแบบไว้
- Function sprintff
 1. รับค่า screen, sym ประเภท string และ ini, res ประเภท float
 2. แสดงค่าทั้งหมดทางจอภาพ ตาม output ที่ออกแบบไว้
- Function print
 1. รับค่า screen, sym ประเภท string และ ini, res, upt ประเภท float
 2. แสดงค่าทั้งหมดทางจอภาพ ตาม output ที่ออกแบบไว้
- Function Aryprint

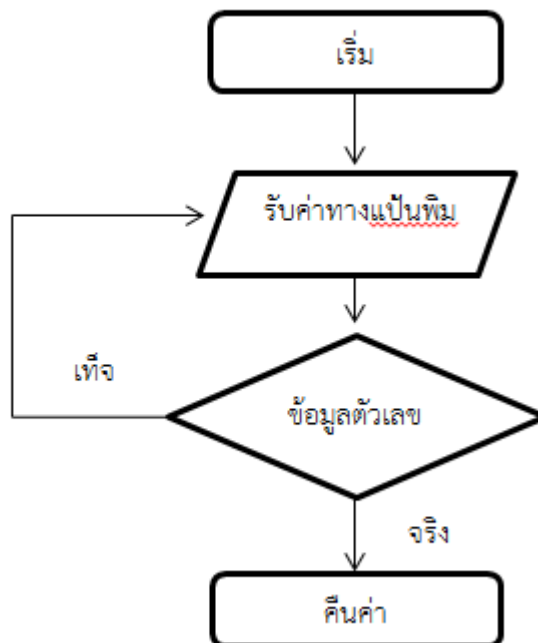
1. รับค่า screen,sym ประเภท string และ ans ประเภท float
2. แสดงค่าทั้งหมดทางจอภาพ ตาม output ที่ออกแบบไว้

3.3 แผนผังลำดับการทำงาน

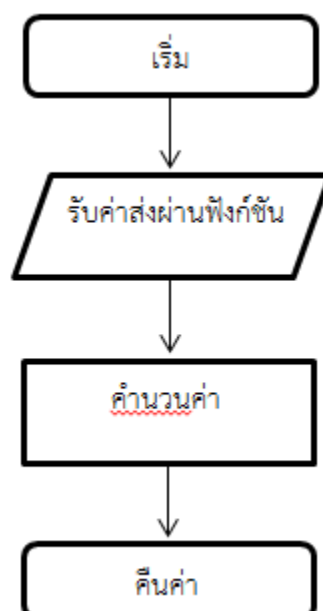
Flow chart: Function(Main) : โปรแกรมหลัก



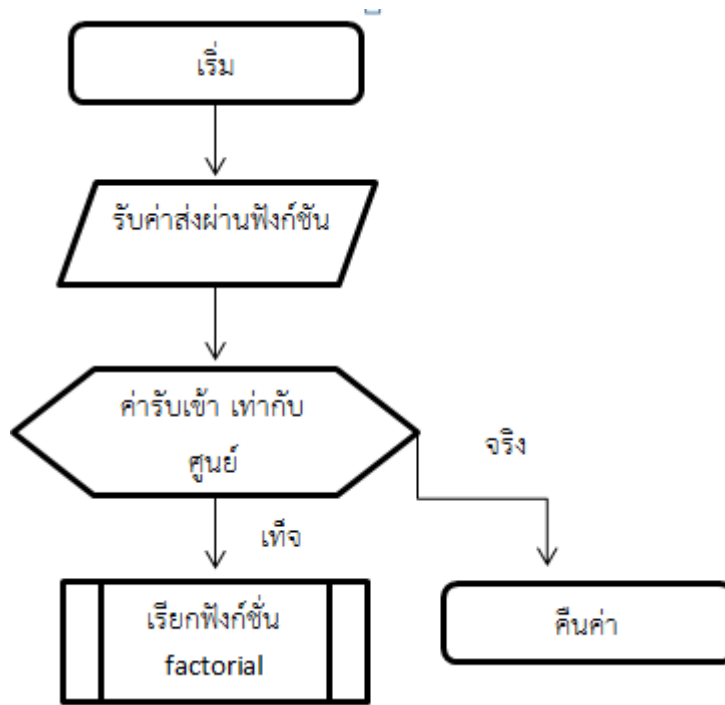
Flow chart: Function(Input,InputAry) : รับค่าตัวเลข



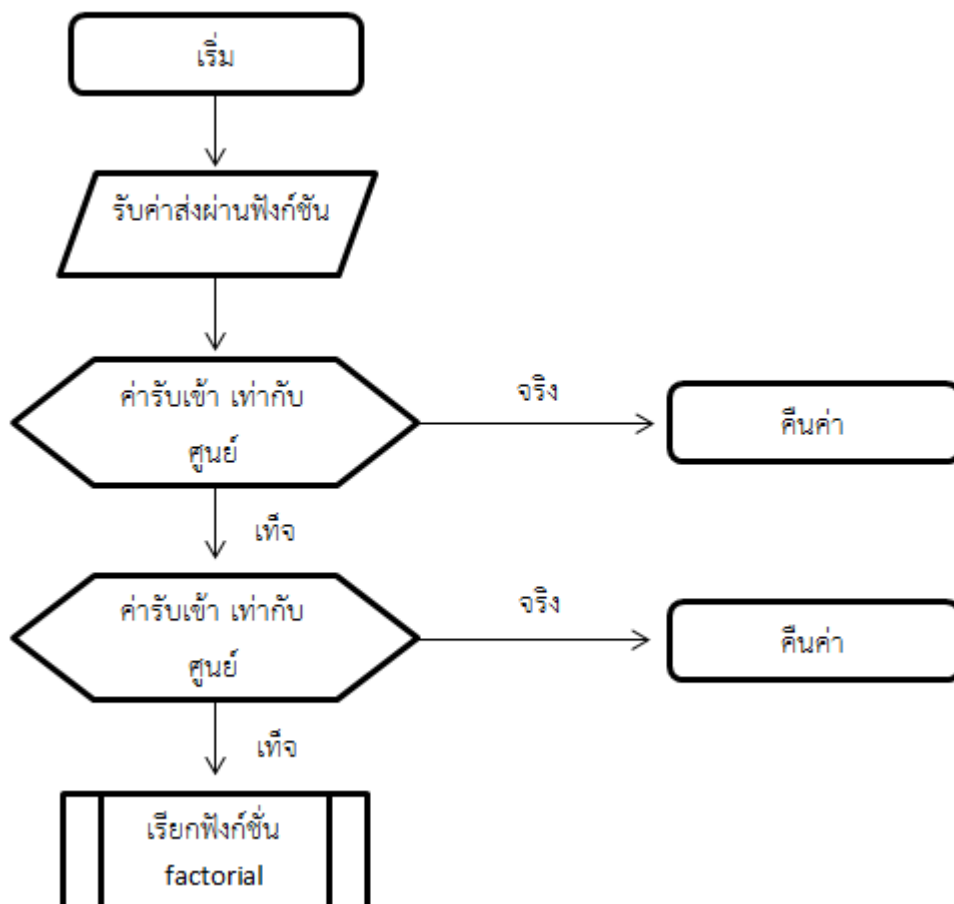
Flow chart: Function(Plus,Minus,Mult,Divind) : คำนวณทั่วไป



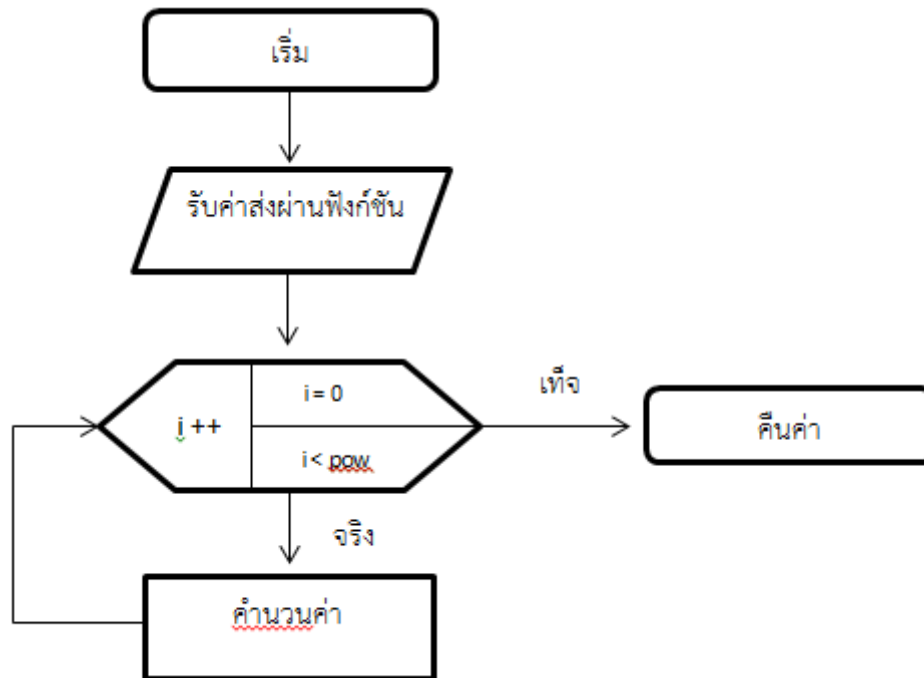
Flow chart: Function(factorial) : Factorial แฟร็กโทเรียล



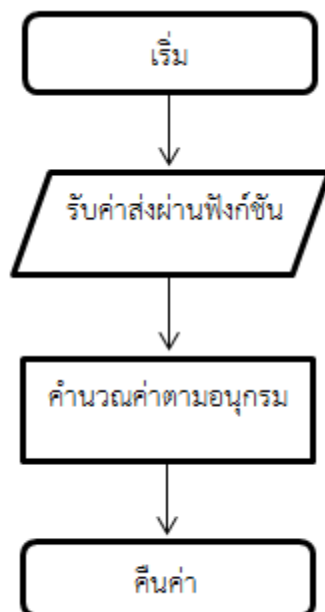
Flow chart: Function(fib) : Fibonacci อนุกรมฟีโบนัชชี



Flow chart: Function(power) : ยกกำลัง



Flow chart: Function(sine,cosine) : ตรีโกณมิติเบื้องต้น



บทที่ 4

ผลการดำเนินงาน

จากแผนการทำงานที่ได้วางไว้ข้างต้น และจากการปฏิบัติตามแผนที่วางไว้ แต่เนื่องจากมีเวลาที่ค่อนข้างจำกัด จึงอาจมีบางอย่างที่ทำให้โครงการไม่เป็นไปตามที่คาดไว้

4.1 โปรแกรมคำนวณทางคณิตศาสตร์

4.1.1 หน้าจอเริ่มต้นและเมนู

```
=====
Menu
=====
1.Regular Calculator
2.Scientific Calculator
3.Accountant Calculator
4.Read Help and Notice
0.Exit
Select Menu: 
```

รูปที่ 4-1 หน้าจอเมื่อเข้าสู่โปรแกรม

ในการเลือกเมื่อนั้น สามารถเลือกได้โดยการใส่ตัวเลขเมนูที่ต้องการ เลือกฟังก์ชันที่อยากจะใช้งาน เช่น เครื่องคิดเลขทั่วไป วิทยาศาสตร์ เป็นต้น

4.1.2 เลือกการคำนวณที่อยากให้กระทำเช่น ในเครื่องคิดเลขวิทยาศาสตร์

```

=====
Scientific Calculator Menu
=====
1.Power function (x^y)
2.Factorial Series (x!)
3.Fibonacci Series
4.Sine (Sin x)
5.Cosine (cos x)
6.Tangent (Tan x)
7.Cosec (cosec x)
8.Sec (sec x)
9.Cot (cot x)
0.Back
Select Menu: █

```

รูปที่ 4-2 หน้าจอเมื่อเลือกเมนู 2 เข้าสู่เครื่องคิดเลขวิทยาศาสตร์

4.1.1 ตัวอย่างการคำนวณใน เครื่องคิดเลขทางสถิติ

```

=====
Accountant Calculator Menu
=====
Please set value first

Enter number of term: 5
Enter value terms[1]: 1
Enter value terms[2]: 2
Enter value terms[3]: 3
Enter value terms[4]: 3
Enter value terms[5]: 5 █

```

รูปที่ 4-3 หน้าจอเมื่อรับค่าไปคำนวณ

4.1.2 แสดง ค่าทางสถิติต่างๆ เช่นค่าฐานนิยม

Mode is: 3.0000

=====

Accountant Calculator Menu

=====

- 1.Show max
 - 2.Show min
 - 3.Show x-bar
 - 4.Show range
 - 5.Show Med
 - 6.Show Mode
 - 7.Show value(sort)
 - 8.Set new value
 - 0.Back
- Select Menu:

รูปที่ 4-4 หน้าจอเมื่อเลือกแสดงค่าฐานนิยม

บทที่ 5

สรุปผลการดำเนินโครงการ

5.1 สรุปผลการดำเนินโครงการ

โปรแกรมการคำนวณทางคณิตศาสตร์ เป็นโปรแกรมที่จัดทำเพื่อสะดวกแก่ผู้ที่สนใจ คณิตศาสตร์และเป็นการนำความรู้ที่ได้เรียนไปจากรายวิชาหลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์ มาใช้ให้เกิดประโยชน์ แต่หากจะดูกันถึงแก่นแท้แล้วยังต้องมีการพัฒนาอีกมากทีเดียว จึงจะสามารถนำไปใช้งานได้จริงอย่างมีประสิทธิภาพ เนื่องจากโปรแกรมนั้นยังมีจุดบกพร่อง แต่ถือว่าโปรแกรมสามารถนำไปใช้งานและพัฒนาต่อยอดได้ทีเดียว

5.2 ประโยชน์ที่ได้รับ

- เป็นการนำความรู้ตลอดภาคการศึกษามาใช้ในการสร้างซอฟต์แวร์จริง
- เกิดการคิดวิเคราะห์ด้วยตนเอง ให้เกิดกระบวนการคิด และแก้ปัญหาด้วยตนเอง
- มีความรู้เกี่ยวกับเครื่องมือภาษา C มากยิ่งขึ้น

5.3 ข้อจำกัด

- บางการคำนวณ จะสามารถทำได้แค่ประมาณค่าเท่านั้น เนื่องจากต้องการความรู้ทางคณิตศาสตร์ และ การโปรแกรมภาษาซี มากกว่านี้
- ถ้าเป็นการรับค่า จะทำได้ทีละขั้นตอนการคำนวณเท่านั้น

ภาคผนวก

Head : se016prj.h

```

/*Program name: se016prj.h
  Student: 57160016 Thammarat kerdlumjiak
  Section: 01 */
#include<stdio.h>
#include<string.h>
#define PI 3.1415926535897932384626433832795

FILE*help;

/*=====Function+Declaration=====
=====*/

float input(char*inname);
float inputAry(char*inname,int num);
float plus(float adder,float addin);
float minus(float miner,float minin);
float mult(float multer,float multin);
float divind(float divider,float dividin);
int factorial(int term);
int fib(int term);
float power(float base,int pow);
float sine(float radius);
float cosine(float radius);
int spprint(char*screen,char*sym,int ini,int res);
float spprintf(char*screen,char*sym,float ini,float res);
float print(char*screen,char*sym,float ini,float upt,float res);
float Aryprint(char*screen,float ans);

```

Main function : se016prj.c

```
/*Program name: se016prj.c
   Student: 57160016 Thammarat kerdlumjiak
   Section: 01 */
#include"se016prj.h"
#define PI 3.1415926535897932384626433832795

int main(int argc,char*argv[])
{
    float a,b,c;
    float r1,r2;
    char sym;
    int menu,rmenu,smenu;

    //-----main menu calculator-----
    ----//
    do{
        printf("\n=====\\n");
        printf("Menu\\n");
        printf("=====\\n");
        printf("1.Regular Calculator\\n");
        printf("2.Scientific Calculator\\n");
```

```

printf("3.Acountant Calculator\n");
printf("4.Read Help and Notice\n");
printf("0.Exit\n");
menu = input("Select Menu: "); //input main menu
system("clear");

//-----regular calculator-----
//
if(menu==1){

do{
    printf("\n===== \n");
    printf("Regular Calculator Menu\n");
    printf("===== \n");
    printf("1.PLUS\n");
    printf("2.MINUS\n");
    printf("3.MULTIPLY\n");
    printf("4.DIVIDE\n");
    printf("0.BACK\n");
    rmenu = input("Select Menu: "); //input regular menu
    system("clear");

//STATEMENT
    if(rmenu==1){
        a=input("Enter value 1 st: ");
        b=input("Enter value 2 nd: ");
        c=plus(a,b);
        printf("\n");
        print("sum of","plus",a,b,c);
    }
}
}

```

```
}//plus a+b
```

```
if(rmenu==2){
```

```
    a=input("Enter value 1 st: ");
```

```
    b=input("Enter value 2 nd: ");
```

```
    c=minus(a,b);
```

```
    printf("\n");
```

```
    print("result of","minus",a,b,c);
```

```
}//minus a-b
```

```
if(rmenu==3){
```

```
    a=input("Enter value 1 st: ");
```

```
    b=input("Enter value 2 nd: ");
```

```
    c=mult(a,b);
```

```
    printf("\n");
```

```
    print("result of","x",a,b,c);
```

```
}//multiply a*b
```

```
if(rmenu==4){
```

```
    a=input("Enter value 1 st: ");
```

```
    b=input("Enter value 2 nd: ");
```

```
    c=divind(a,b);
```

```
    printf("\n");
```

```
    print("result","/",a,b,c);
```

```
}//divind a/b
```

```
}while(rmenu!=0);
```

```
}//END REGULAR
```

```
//-----scientific calculator-----
--//
if(menu==2){

    do{

        printf("\n=====\\n");
        printf("Scientific Calculator Menu\\n");
        printf("=====\\n");
        printf("1.Power function (x^y)\\n");
        printf("2.Factorial Series (x!)\\n");
        printf("3.Fibonacci Series \\n");
        printf("4.Sine (Sin x)\\n");
        printf("5.Cosine (cos x)\\n");
        printf("6.Tangent (Tan x)\\n");
        printf("7.Cosec (cosec x)\\n");
        printf("8.Sec (sec x)\\n");
        printf("9.Cot (cot x)\\n");
        printf("0.Back\\n");

        smenu = input("Select Menu: "); //input scientific menu
        system("clear");

    if(smenu==1){

        a=input("Enter base(x): ");
        b=input("Enter power(y): ");
        c=power(a,b);
        printf("\\n");
        print("result of", "^", a,b,c);

    } //power a^b

    if(smenu==2){
```

```

        a=input("Enter numbers of term: ");
        c=factorial(a);
        printf("\n");
        sprintf("Factorial of", "%d", a, c);
    }//Factorial x!

```

```

if(smenu==3){
    a=input("Enter numbers of term: ");
    c=fib(a);
    printf("\n");
    sprintf("Fibonacci of", "%d", a, c);
} //Fibonacci n term

```

```

if(smenu==4){
    a=input("Enter your value: ");
    c=sine(a);
    printf("\n");
    sprintf("Sine of", "%d", a, c);
} //Sine x

```

```

if(smenu==5){
    a=input("Enter your value: ");
    c=cosine(a);
    printf("\n");
    sprintf("Cosine of", "%d", a, c);
} //cos x

```

```

if(smenu==6){
    a=input("Enter your value: ");
    r1=sine(a);

```

```

        r2=cosine(a);
        c=r1/r2;
        printf("\n");
        sprintf("Tangent of", " ",a,c);
    }//tan x

    if(smenu==7){
        a=input("Enter your value: ");
        c=sine(a);
        printf("\n");
        sprintf("Cosec of", " ",a,1/c);
    }//cosec x

    if(smenu==5){
        a=input("Enter your value: ");
        c=cosine(a);
        printf("\n");
        sprintf("Sec of", " ",a,1/c);
    }//sec x

    if(smenu==8){
        a=input("Enter your value: ");
        r1=sine(a);
        r2=cosine(a);
        c=r1/r2;
        printf("\n");
        sprintf("Cot of", " ",a,1/c);
    }//cot x

}while(smenu!=0);

```

```
}//END SCIENTIFIC
```

```
//-----Accountat Calculator-----  
-----//
```

```
if(menu==3){
```

```
    warp:
```

```
        system("clear");  
        printf("\n=====\\n");  
        printf("Accountant Calculator Menu\\n");  
        printf("=====\\n");  
        printf("Please set value first\\n");  
        printf("\\n");  
        //Head text
```

```
    int n,i,j,amenu;  
    float rest,max,min,x_bar=0,sum=0;  
    float med,mod,count;  
    int temp;
```

```
        n=input("Enter number of term: ");  
        //Numbers of N
```

```
    float set[n];  
    int numtemp[n];
```

```
    for(i=0;i<n;i++){  
        set[i]=inputAry("Enter value terms",i);  
    } //Get value to array
```

```

        for(i=0;i<n;i++){
            for(j=0;j<=i;j++){
                if(set[j]>set[i]){
                    rest=set[j];
                    set[j]=set[i];
                    set[i]=rest;
                }
            }
        }
    }//Numbers Sort

//=====EQUALTION=====//
//=====>

//Max&Min
min = set[0];
max = set[n-1];
//=====>

//X-bars
for(i=0;i<n;i++){
    sum+=set[i];
}
//sum value
x_bar=(sum/n);
//average value to x-bar
//=====>

//Med
if((n%2)!=0){
    med=set[((n+1)/2)-1];
}

```



```

} //Set value of med
else{
    med=(set[((n+1)/2)]+set[((n+1)/2)-1])/2;
} //Set value when med%2 is not 0

//=====>

//Mode
for(i=0;i<n;i++){
    numtemp[i]=0;
}
for(i=0;i<n;i++){
    temp=set[i];
    for(j=i;j<n;j++){
        if(set[j]==temp){
            numtemp[i]++;
        }
    }
}
} //Check in any lower to highter for count same numbers

temp=numtemp[0];
for(i=1;i<n;i++){
    if(numtemp[i]>temp){
        temp = numtemp[i];
    }
}
} //Check while highter any position is more than lower && set highter value to
temp

if(i==999){ //Always False If

```

```

warp1://Warp form menu
j=0;

for(i=0;i<n;i++){
    if(numtemp[i]==temp){
        j++;
    }
}
} //Count j for check Mod value

if(j==1){
    for(i=0;i<n;i++){
        if(numtemp[i]==temp){
            printf("%.4f\n",set[i]);
        }
    }
} //True Mod is set[i]:

else{
    printf("Not have mode value\n");
} //Else Mod > 1 value:

}

//=====>
//Accountant cal Menu
do{
    printf("\n===== \n");
    printf("Accountant Calculator Menu\n");
    printf("===== \n");
    printf("1.Show max\n");

```

```

printf("2.Show min\n");
printf("3.Show x-bar\n");
printf("4.Show range\n");
printf("5.Show Med\n");
printf("6.Show Mode\n");
printf("7.Show value(sort)\n");
printf("8.Set new value\n");
printf("0.Back\n");
amenu = input("Select Menu: "); //input accountant menu
system("clear");

//=====>

if(amenu==1){
    printf("\n");
    Aryprint("Max is",max);
}

if(amenu==2){
    printf("\n");
    Aryprint("Min is",min);
}

if(amenu==3){
    printf("\n");
    Aryprint("X-bar is",x_bar);
}

if(amenu==4){
    printf("\n");
    Aryprint("Range is",max-min);
}

```

```
}

if(amenu==5){
    printf("\n");
    Aryprint("Med is",med);
}

if(amenu==6){
    printf("\n");
    printf("Mode is: ");
    goto warp1;
}

if(amenu==7){
    printf("\n");
    printf("Set of number is(sort): ");
    for(i=0;i<n;i++){
        printf("%.3f ",set[i]);
    }
}

if(amenu==8){
    goto warp;
}

}while(amenu!=0);
}

if(menu==4){
    char text;
```

```
        help = fopen("User_helping.txt","r");

        while((text=fgetc(help))!=EOF){
            fprintf(stdout,"%c",text);
        }

        fclose(help);
    }
}while(menu!=0);

return 0;
}
```

Sub function : se016lib.c

```

/*Program name: se016lib.c
Student: 57160016 Thammarat kerdlumjiak
Section: 01 */
#include"se016prj.h"
#define PI 3.1415926535897932384626433832795

float input(char*inname)
{
    float key;
    char temp[9999];

    do{
        printf("%s",inname);
        if(!scanf("%f",&key)){
            scanf("%s",&temp);//recieve temp
            printf("Please try again!\n");
        }
        else{
            return key;
        }
    }while(1);
}

float inputAry(char*inname,int num)
{
    float key;
    char temp[9999];

```

```
do{
printf("%s[%d]: ",inname,num+1);
if(!scanf("%f",&key)){
    scanf("%s",&temp);
    printf("Please try again!\n");
}
else{
    return key;
}
}while(1);
}
```

```
float plus(float adder,float addin)
{
    float result;
    result=adder+addin;
    return result;
}
```

```
float minus(float miner,float minin)
{
    float result;
    result=miner-minin;
    return result;
}
```

```
float mult(float multer,float multin)
{
    float result;
    result=multer*multin;
```

```
        return result;
    }

float divind(float divider,float dividin)
{
    float result;
    result=divider/dividin;
    return result;
}

//-----
int factorial(int term)
{
    if(term==0){
        return 1;
    }

    return term*factorial(term-1);
}

int fib(int term)
{
    if(term==1){
        return 0;
    }

    if(term==2){
        return 1;
    }
}
```



```
    }

    return fib(term-1)+fib(term-2);
}

float power(float base,int pow)
{
    int i;
    float mem=1;

    for(i=0;i<pow;i++){
        mem*=base;
    }

    return mem;
}

float sine(float radius)
{
    float val,sin;

    val=radius*(PI/180);
    sin=val-(power(val,3)/factorial(3))+(power(val,5)/factorial(5))-
(power(val,7)/factorial(7));

    return sin;
}

float cosine(float radius)
{

```

```

float val,cos;

        val=radius*(PI/180);
        cos=1-(power(val,2)/factorial(2))+(power(val,4)/factorial(4))-
(power(val,6)/factorial(6));

        return cos;
}
//-----
int spprint(char*screen,char*sym,int ini,int res)
{
        printf("%s %d%s = %d\n",screen,ini,sym,res);

return 0;
}

float spprintf(char*screen,char*sym,float ini,float res)
{
        printf("%s %.4f%s = %.4f\n",screen,ini,sym,res);

return 0;
}

float print(char*screen,char*sym,float ini,float upt,float res)
{
        printf("%s %.4f %s %.4f = %.4f\n",screen,ini,sym,upt,res);

return 0;
}
float Aryprint(char*screen,float ans)

```

```
{  
    printf("%s : %.4f\n",screen,ans);  
  
    return 0;  
}
```