



Software Park Thailand
</Code Camp>

Outline

- ❑ วิทยาการคอมพิวเตอร์คืออะไร
- ❑ ไบนารี
- ❑ การแสดงผลข้อมูล
- ❑ อัลกอริทึม
- ❑ รหัสเทียม (Pseudo code)
- ❑ สแครช (Scratch)
- ❑ คำจำกัดความของเทคโนโลยี
- ❑ ทฤษฎี และการออกแบบ อัลกอริทึม
- ❑ สถาปัตยกรรมคอมพิวเตอร์

Objective : ทฤษฎี และการออกแบบอัลกอริทึม

- ❑ โครงสร้างของข้อมูล
- ❑ ประเภทของข้อมูล
- ❑ ADT (Abstract data type)
- ❑ Algorithm
- ❑ Flow chart

ทบทวนโครงสร้างของข้อมูล

1. โครงสร้างข้อมูล แบบ**อาร์เรย์**
2. โครงสร้างข้อมูล แบบ**ลิงค์ลิส**
3. โครงสร้างข้อมูล แบบ**สแต็ก**
4. โครงสร้างข้อมูล แบบ**คิว**



1. โครงสร้างข้อมูล แบบ**อาร์เรย์**

โครงสร้างข้อมูลพื้นฐานที่สุดได้แก่ อาร์เรย์ (arrays) หรือ แถวลำดับ ซึ่งเป็นพื้นที่หน่วยความจำที่ระบุขนาดหรือจำนวนหน่วยที่ใช้เก็บข้อมูลไว้ตั้งแต่ต้นเมื่อตอนประกาศตัวแปร อาร์เรย์จะใช้พื้นที่หน่วยความจำที่อยู่เรียงติดกันและใช้เก็บข้อมูลชนิดเดียวกัน พื้นที่หน่วยความจำที่ ประกาศใช้เป็นอาร์เรย์ นี้จะถูกจองเพื่อการใช้งานอยู่ตลอดเวลา หากใช้งานเพียงบางส่วน ส่วนที่เหลือก็จะไม่มีผู้ใดใช้ได้ ทำให้การใช้โครงสร้างข้อมูลแบบอาร์เรย์ ไม่มีประสิทธิภาพเท่าที่ควร

1. โครงสร้างข้อมูล แบบอาร์เรย์

มีอาร์เรย์ 2 ประเภทที่มัก จะพบเห็นและใช้กันทั่วไปคือ เวกเตอร์ (vector) ซึ่งเป็นโครงสร้างอาร์เรย์ แบบ 1 มิติ ($k=1$) และเมตริกซ์ (matrix) ซึ่งเป็นโครงสร้าง แบบอาร์เรย์ แบบ 2 มิติ ($k=2$)

A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
a	b	c	d	e	f	g	h	i	j

2.)

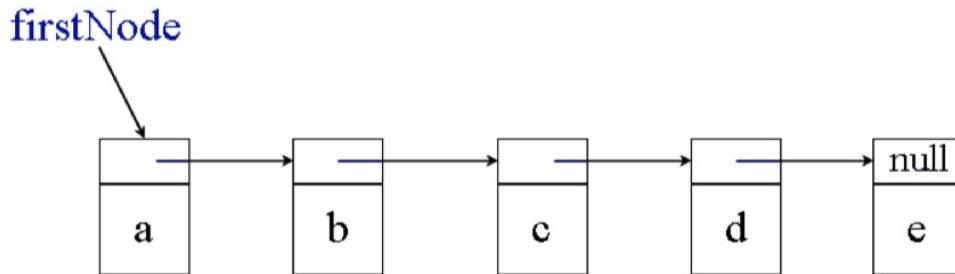
โครงสร้างข้อมูล แบบลิงค์ลิสต์

การเก็บข้อมูลในลักษณะที่เป็นบัญชีรายการหรือลิสต์สามารถเก็บได้โดยใช้โครงสร้างแบบอาร์เรย์ซึ่งจะเก็บข้อมูลในพื้นที่หน่วยความจำที่ต่อเนื่องกัน และเราสามารถเก็บข้อมูลในลักษณะที่เป็นชุดข้อมูลเชื่อมโยงต่อเนื่องกันได้ โดยใช้โครงสร้างข้อมูลที่เรียกว่า ลิงค์ลิสต์ (linked list)



ลิงค์ลิสต์เดี่ยว

ลิงค์ลิสต์เดี่ยว หมายถึง ลิงค์ลิสต์ที่แต่ละโหนดที่ใช้เก็บข้อมูลแสดงถึงลำดับก่อนหลัง อย่างชัดเจนโดยใช้พอยน์เตอร์บอกว่า ข้อมูลหรือโหนดตัวต่อไปอยู่ที่ใด

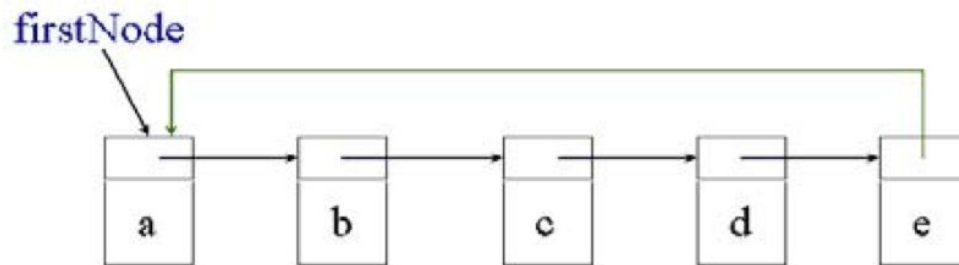


size = number of elements



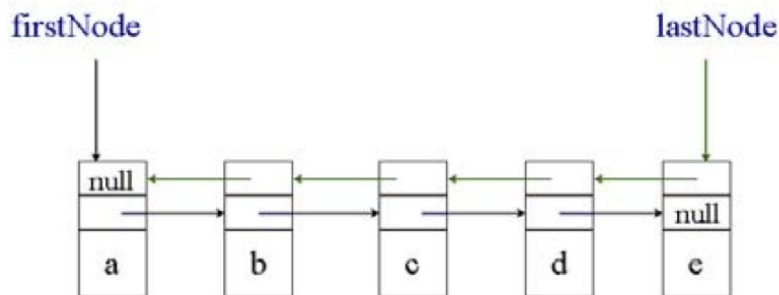
ลิงค์ลิสต์เดี่ยวแบบวงกลม

ลิงค์ลิสต์เดี่ยวแบบวงกลม คล้ายกับลิงค์ลิสต์เดี่ยวแตกต่างกัน ที่พอยน์เตอร์ ของโหนดสุดท้ายของลิงค์ลิสต์เดี่ยวแบบวงกลมที่เก็บแอดเดรสของโหนดแรกแทนที่จะ เป็น null



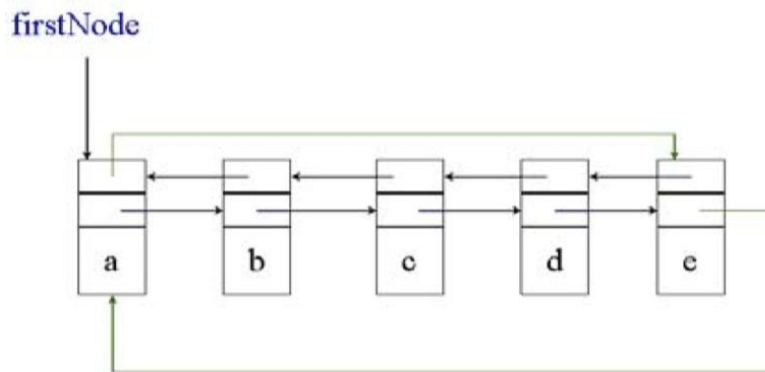
ลิงค์ลิสต์คู่

ลิงค์ลิสต์คู่ เป็นลิงค์ลิสต์ที่แต่ละโหนดมีสองพอยน์เตอร์ คือพอยน์เตอร์ที่ชี้ไปยังโหนดถัดไป และพอยน์เตอร์ที่ชี้ไปยังโหนดก่อนหน้าของโหนดนั้น แต่ละโหนดของลิงค์ลิสต์คู่จึงมี 3 ช่อง ใช้สำหรับเก็บข้อมูลหนึ่งช่อง และอีกสองช่องใช้เก็บค่าพอยน์เตอร์ (ที่ชี้ไปยังโหนดถัดไป และโหนดก่อนหน้า)



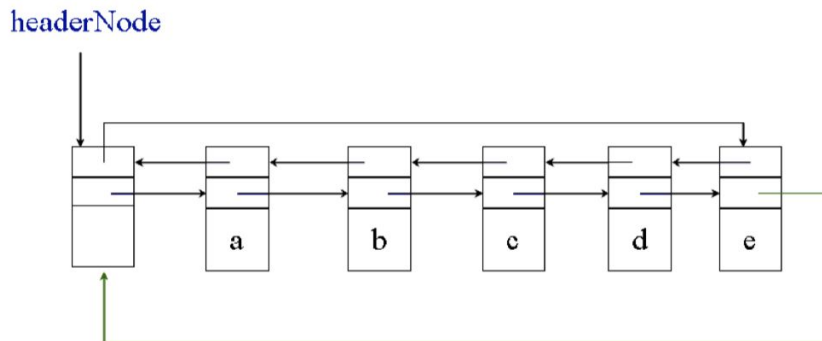
ลิงค์ลิสต์คู่แบบวงกลม

ลิงค์ลิสต์คู่แบบวงกลม มีลักษณะคล้ายกับลิงค์ลิสต์เดี่ยวแบบวงกลมที่มีการกำหนดให้ พอยน์เตอร์ตัวหนึ่งที่โหนดสุดท้ายชี้ไปยังโหนดแรก และมีพอยน์เตอร์อีกตัวหนึ่งที่ โหนดแรกชี้ไปที่โหนดสุดท้าย



ลิงค์ลิสต์คู่แบบวงกลมที่มีหัวโหนด

ลิงค์ลิสต์คู่แบบวงกลมที่มีหัวโหนด ลิงค์ลิสต์อาจมีโหนดหัว (headernode) เพื่อใช้เก็บข้อมูลเกี่ยวกับลิงค์ลิสต์ เช่น จำนวนโหนดที่มีอยู่ในลิงค์ลิสต์ และพอยน์เตอร์สำหรับชี้ไปยังโหนดถัดไป (สำหรับ กรณีลิงค์ลิสต์เดี่ยว) หรือพอยน์เตอร์ ที่ชี้ไปยังโหนดถัดไป และพอยน์เตอร์ที่ชี้ไปยังโหนดก่อนหน้า (สำหรับกรณี ลิงค์ลิสต์คู่)

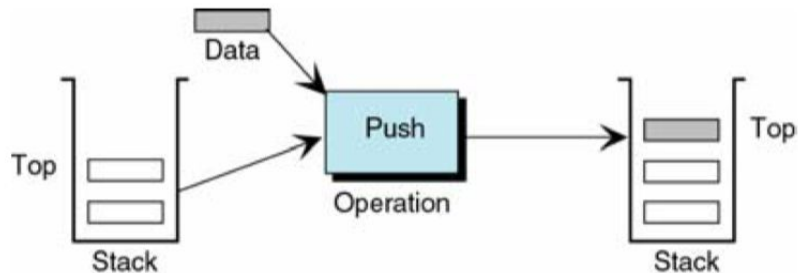


3.) โครงสร้างข้อมูล แบบสแต็ก

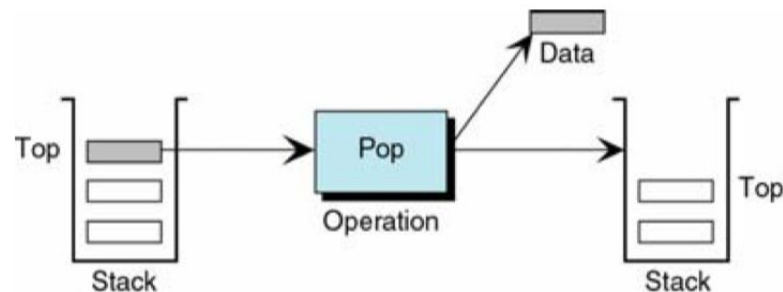
โครงสร้างข้อมูลแบบสแต็ก (stack) เป็นโครงสร้างการเก็บข้อมูลแบบเรียงทับซ้อน ที่ใช้ นโยบายการจัดการข้อมูลแบบเข้าก่อน ออกทีหลังหรือ LIFO (last-in-first-out) คำสั่งพื้นฐานที่ เกี่ยวข้องกับข้อมูลที่มีโครงสร้างประเภทนี้ ได้แก่ คำสั่ง PUSH, POP, และ TOP



คำสั่ง PUSH และคำสั่ง POP



(a) คำสั่ง PUSH เพิ่มข้อมูลในสแต็ก

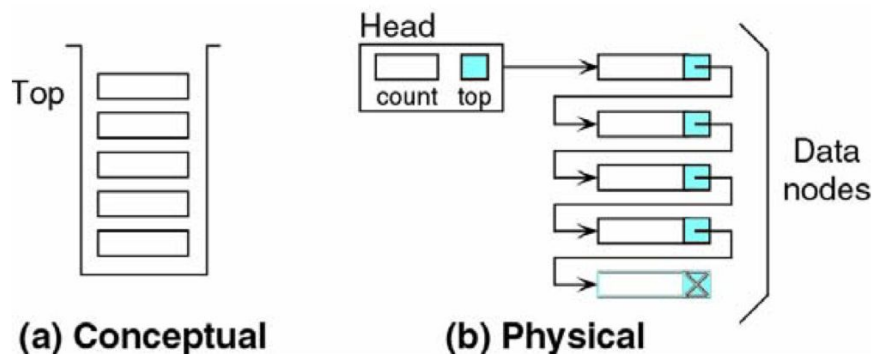


(b) คำสั่ง POP นำเอาข้อมูลออกจากสแต็ก



สแต็ก

สแต็ก สามารถแทนด้วยโครงสร้างข้อมูล แบบอาเรย์หรือลิงค์ลิสต์ ถ้าแทนสแต็กด้วยอาเรย์ การเก็บข้อมูลก็ต้องเป็นไปตามข้อจำกัดของโครงสร้างข้อมูลแบบอาเรย์ ถ้าแทนสแต็กด้วยลิงค์ลิสต์ ก็จะมีคามยืดหยุ่น



4.)

โครงสร้างข้อมูล แบบคิว

โครงสร้างข้อมูลแบบคิว (queue) เป็นโครงสร้างการเก็บข้อมูลที่ใช้นโยบายการจัดการข้อมูลแบบเข้าก่อน ออกก่อนหรือ FIFO (first-in-first-out)

คำสั่งพื้นฐานที่เกี่ยวข้องกับ ข้อมูลที่มี โครงสร้างประเภทนี้ได้แก่ คำสั่ง ENQUEUE, DEQUEUE, FRONT, REAR

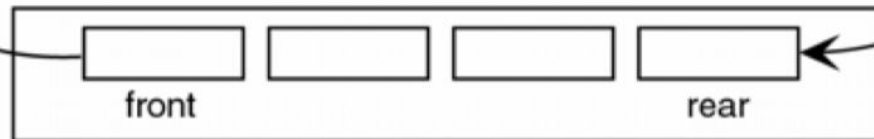


Banks'R'Us



(a) A queue (line) of people

Remove
(dequeue)



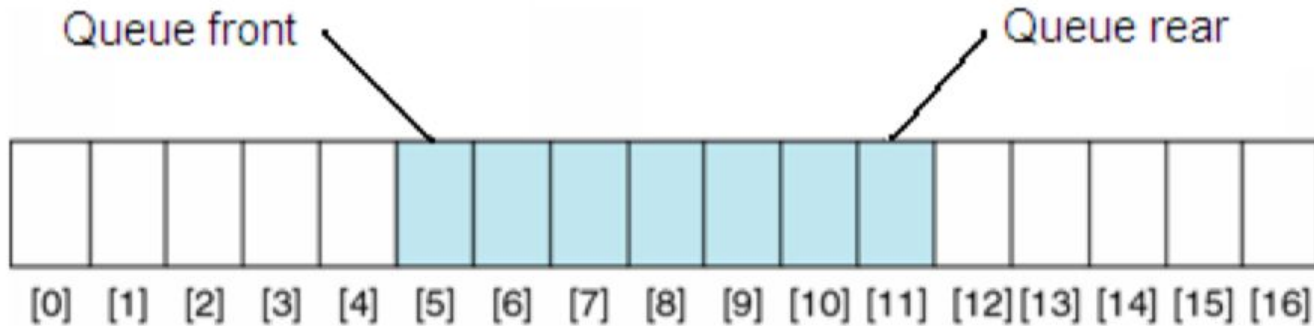
Insert
(enqueue)

(b) A computer queue



การแทนโครงสร้างข้อมูลแบบคิว

การแทนโครงสร้างข้อมูลแบบคิว สามารถทำได้เช่นเดียวกับสแต็ก คือ แทนด้วยอาเรย์ หรือ ลิงค์ลิสต์ก็ได้



หลังจากทบทวนโครงสร้างข้อมูล
แล้ว มาเริ่มอัลกอริทึมกันเลย



ส่วนประกอบของประเภทข้อมูล

ส่วนประกอบของประเภทข้อมูล



1 Bit มีทั้งหมด 8 Byte

Type	Values	Operations
Integer	$-\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty$	$+, -, *, /, ++, --, \dots$
Floating point	$-\infty, \dots, 0.0, \dots, \infty$	$+, -, *, /, \dots$
character	$\backslash 0, \dots, 'A', 'B', \dots, 'a', 'b', \dots$	$<, >, \dots$



ADT (Abstract data type)

ADT (Abstract data type) หมายถึงประเภทข้อมูล ซึ่งแสดงถึงระบบ การจัดการข้อมูล โดยแสดงถึงบริการ และกฎเกณฑ์ในการจัดการข้อมูล นั้น ๆ แต่ไม่ได้กล่าวถึงวิธีการสร้างบริการ ต่างๆ (คือ การนำมาใช้ ไม่ใช่การศึกษาวิธีสร้าง)

หลักการของ abstraction

- We know what a data type can do
- How it is done is hidden

ลักษณะของ ADT แบ่งตามความสัมพันธ์ในข้อมูล

- ข้อมูลเชิงเดี่ยว (Atomic data)
- ข้อมูลประกอบ (Composite data)

ข้อมูลเชิงเดี่ยว (Atomic Data)

ข้อมูลเชิงเดี่ยว (Atomic Data) คือ ข้อมูลที่ประกอบด้วยค่าเดียวที่ไม่ สามารถแบ่ง
ส่วนข้อมูลนี้ออกไปเพื่อสื่อ ความหมายได้อีก เช่น เลข จำนวนเต็ม อายุ เพศ เป็นต้น

ข้อมูลประกอบ (Composit Data)

ข้อมูลประกอบ (Composit Data) คือ ข้อมูลที่สามารถแตกออกเป็น ฟิลด์ย่อยๆ ได้ และสิ่งที่แตกออกไปสามารถสื่อความหมายได้ เช่น เบอร์โทรศัพท์ หมายเลข ครัวภัณฑ์ รหัสนักศึกษา เป็นต้น

ตัวอย่างการเขียนอัลกอริทึม

ในรูปคำสั่งเทียม (Pseudo code) คือมีลักษณะเป็น การเขียนคำสั่งต่างๆ ที่ทำให้
ผู้เขียนโปรแกรมสามารถเข้าใจง่ายซึ่งคำสั่งที่ใช้จะเป็นคำพูดธรรมดาไม่ยึดหลัก
ไวยากรณ์ของภาษาคอมพิวเตอร์

ตัวอย่าง :

ตัวอย่างที่ 1 การหาผลรวมของตัวเลข N จำนวน

Program : Sum

รับค่าจำนวนตัวเลขทั้งหมดเก็บไว้ที่ตัวแปร N

กำหนดให้ตัวแปรเก็บตัวนับเริ่มต้น $I = 1$, เก็บผลรวมเริ่มต้น $Sum = 0$

Repeat

รับค่าข้อมูลเก็บไว้ที่ X

นำค่าข้อมูลที่อ่านได้บวกสะสมไว้ที่ตัวแปรเก็บผลรวม Sum

เพิ่มค่าตัวนับ I ขึ้นอีก 1

Until เป็นเลขตัวสุดท้ายจริง ($I > N$)

แสดงตัวแปรที่เก็บผลรวมสะสม Sum

End Program

ตัวอย่าง :

ตัวอย่างที่ 2 หาค่าสูงสุด และ ต่ำสุดจากตัวเลขทั้งหมด N จำนวน

Program : Max_Min

รับค่าจำนวนตัวเลขทั้งหมดเก็บไว้ในตัวแปร N

กำหนดให้ตัวแปรเก็บตัวนับเริ่มต้น $I = 1$

อ่านเลขตัวแรกเก็บไว้ในตัวแปร X

กำหนดให้ตัวแปร X เป็นทั้งค่ามากที่สุดและน้อยที่สุด เก็บไว้ในตัวแปร Max
และ Min

Repeat

อ่านเลขตัวถัดไป เก็บไว้ใน X และเพิ่มค่าตัวนับ I อีก 1

If $X > \text{Max}$ Then

เปลี่ยนค่าใน Max ให้เก็บค่า X แทน

Else

If $X < \text{Min}$ Then

เปลี่ยนค่าใน Min ให้เก็บค่า X แทน

End if

End if

ให้ค่า $\text{Max} = \text{Max}$, ค่า $\text{Min} = \text{Min}$

Until $I > n$











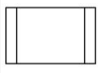
แสดง ตัวเลขในตัวแปร Max และ Min

End Program


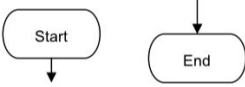











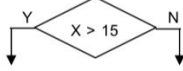




ทฤษฎี และการออกแบบอัลกอริทึม

โฟลว์ชาร์ต (Flow Chart) หมายถึง วิธีการออกแบบโปรแกรมอย่างเป็นขั้นตอน และมีเหตุผลผลโดยการใช้สัญลักษณ์ และตัวอักษรประกอบการอธิบายในแต่ละขั้นตอน เพื่อให้ผู้เขียนโปรแกรมและผู้ที่สนใจทั่วไปสามารถเข้าใจความหมายได้

สัญลักษณ์ที่ใช้ใน การเขียน Flow chart (1)

สัญลักษณ์	ชื่อเรียก	หน้าที่
	Terminal / interrupt (เริ่มต้น / สิ้นสุด)	แสดงจุดเริ่มต้นหรือสิ้นสุดของผังงาน (ในผังงานจะมีจุดเริ่มต้นและสิ้นสุดการทำงานอย่างละ 1 จุดเท่านั้น)
	Assignment (การกำหนดค่า) Computation (การคำนวณ) Process (การประมวลผล)	สัญลักษณ์ที่ใช้สำหรับแสดงการประมวลผล เช่น การคำนวณ หรือ การกำหนดค่า
	Input (การนำข้อมูลเข้า) Output (การนำข้อมูลออก)	การรับหรือแสดงข้อมูลที่ไม่ได้กำหนดอุปกรณ์ในการรับ หรือแสดงผล
	Input (รับข้อมูลเข้าทางแป้นพิมพ์)	การรับข้อมูลเข้าทางแป้นพิมพ์ (Keyboard)
	Display	การแสดงค่าข้อมูลออกทางจอภาพ
	Document	การแสดงค่าข้อมูลออกทางเครื่องพิมพ์
	Decision	สัญลักษณ์ที่ใช้สำหรับการตัดสินใจ
	Flowchart Connector	จุดเชื่อมต่อผังงานในหน้าเดียวกัน
	Flowchart Off- Page Connector	จุดเชื่อมต่อผังงานคนละหน้า
	ถูกสร	ถูกสรแสดงทิศทางการไหลของผังงาน
	Call Procedure	การเรียกใช้โปรแกรมย่อย

สัญลักษณ์ที่ใช้ในการเขียน Flow chart (2)

สัญลักษณ์	การใช้งาน	คำอธิบาย
		แสดงจุดเริ่มต้นหรือสิ้นสุดของผังงาน (ในผังงานจะมีจุดเริ่มต้นและสิ้นสุดการทำงานอย่างละ 1 จุดเท่านั้น)
		การกำหนดค่า A ให้มีค่าเท่ากับ 5 (ในการกำหนดค่าให้กับตัวแปร จะไม่ใช่เครื่องหมาย = เพราะมีไว้สำหรับการเปรียบเทียบค่า 2 ค่า จะใช้เครื่องหมาย ← แทน)
		การรับค่าข้อมูลมาเก็บไว้ในตัวแปร A และตัวแปร B โดยค่าแรกจะถูกนำไปเก็บไว้ในตัวแปร A และ ค่าถัดไปเก็บไว้ในตัวแปร B
		การรับค่าจากแป้นพิมพ์มาเก็บไว้ในตัวแปร A และตัวแปร B โดยค่าแรกที่รับเข้ามาจะถูกนำไปเก็บไว้ในตัวแปร A และค่าถัดไปจะถูกนำไปเก็บไว้ในตัวแปร B
		การแสดงค่าในตัวแปร ANS ออกทางจอภาพ
		การแสดงค่าในตัวแปร ANS ออกทางเครื่องพิมพ์
		จากภาพจะพิจารณาในตัวแปร X ก่อน ถ้าในตัวแปร X ในขณะนั้นมากกว่า 15 จะทำในขั้นตอนการทำงานในทิศทางของ Y ถ้าในตัวแปร X ไม่มากกว่า 15 จะทำในขั้นตอนการทำงานในทิศทางของ N
		แสดงเครื่องหมายแสดงจุดต่อภายในหน้าเดียวกัน หมายเลข 1 โดยที่ทั้ง 2 จุดจะต้องอยู่หน้าเดียวกัน โดยเสมือนว่าสองจุดนี้เชื่อมต่อกัน
		แสดงเครื่องหมายจุดต่อระหว่างหน้า หมายเลข 1 โดยเครื่องหมายแสดงจุดต่อทั้งสองจุดนั้นจะต้องอยู่คนละหน้ากัน เสมือนว่าทั้งสองจุดนี้เชื่อมต่อกัน

ทฤษฎี และการออกแบบอัลกอริทึม

Flow Chart Template

1. **การกำหนดค่าเริ่มต้น** เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆ ที่จำเป็นบางตัว ได้แก่ ตัวแปรที่ใช้เป็น ตัวนับ หรือตัวแปรที่เป็น ตัวคำนวณผลรวมต่างๆ
2. **การรับข้อมูล** เข้าเป็นการรับข้อมูลนำเข้ามาจากแหล่งข้อมูลต่างๆ เพื่อใช้เป็นส่วนหนึ่ง ของการประมวลผล แล้วนำค่ามาเก็บไว้ในตัวแปรใดๆ ที่กำหนดเอาไว้
3. **การประมวลผล** เป็นการประมวลผลตามที่ได้มีการกำหนด หรือเป็นการคำนวณต่างๆ ซึ่งจะต้องทำที่ละลำดับขั้นตอน และแยกรูปแต่ละรูปออกจากกันให้ชัดเจนด้วย
4. **การแสดงผลลัพธ์** เป็นการแสดงข้อมูลที่ได้จากการคำนวณหรือผลลัพธ์ที่ต้องการหรือค่า จากตัวแปรต่างๆ ซึ่งการแสดงผลลัพธ์นี้มัก จะกระทำหลัง จากการประมวลผลหรือหลังจาก การรับข้อมูล เข้ามาแล้ว

ทฤษฎี และการออกแบบอัลกอริทึม

หลักการเขียนโปรแกรม

โปรแกรมคำสั่ง (Programming Language) โดยทั่วไป หมายถึง ชุด ของคำสั่ง จำนวนหนึ่งที่ถูกจัดเป็นลำดับขั้นตอนของการปฏิบัติงานเพื่อ ที่จะได้ผลลัพธ์ที่ต้องการ อย่างหนึ่ง คำสั่งเหล่านั้นถูกเขียน ในรูปแบบตามข้อกำหนดหรือข้อบังคับของภาษาที่ใช้

ทฤษฎี และการออกแบบอัลกอริทึม

สรุปขั้นตอนของการวิเคราะห์งาน 5 ขั้นตอน ได้ดังนี้ คือ

1. ระบุสิ่งที่โจทย์ต้องการ
2. ออกแบบหน้าจอผลลัพธ์
3. ระบุข้อมูลนำเข้า
4. กำหนดตัวแปรที่ใช้
5. ขั้นตอนการประมวลผล (อัลกอริทึม) หรือโฟลว์ชาร์ต

ตัวอย่าง :

จงหาผลรวมตั้งแต่ 1
จนถึง N

1. สิ่งที่ต้องพิจารณา

ผลรวมของจำนวนตัวเลขตั้งแต่ 1 – N

2. รูปแบบผลลัพธ์

Program	Sum	Number 1 – N
<hr/>		
Input	Top number of N = ?	
	Total Of Number 1 – N = x	
<hr/>		
- Press < any key > To End Program -		

3. ข้อมูลนำเข้า

จำนวนตัวเลขเริ่มต้นที่ 1, 2, 3, ... ไปเรื่อยๆ จนกระทั่งถึง N

4. ตัวแปรที่ใช้

- 4.1 Top_number = ตัวแปรเก็บตัวเลขสูงสุด
- 4.2 Number = ตัวแปรเก็บตัวเลขนำเข้าตั้งแต่ 1 – N
- 4.3 Total = ตัวแปรเก็บผลรวมของจำนวนตัวเลขตั้งแต่ 1 – N

5. ขั้นตอนหรือวิธีการในการประมวลผล (อัลกอริทึม)

1. พิมพ์หัวโปรแกรมการหาผลรวมเลข 1 – N
2. ป้อนค่าตัวเลขสุดท้าย เก็บไว้ที่ตัวแปร Top_number
3. กำหนดค่าตัวแปรเก็บตัวเลขเริ่มต้น Number = 1 และตัวแปรเก็บผลรวมเริ่มต้น Total = 0
4. นำค่าตัวแปร Number บวกสะสมเก็บไว้ที่ตัวแปร Total
5. เพิ่มค่าตัวแปร Number ขึ้นอีก 1
6. เช็คว่า Number > Top_number หรือยัง
 - ถ้าเป็นจริง ไปทำขั้นตอนที่ 7
 - ถ้าเป็นเท็จ กลับไปทำขั้นตอนที่ 4
7. พิมพ์ผลรวมสะสมที่คำนวณได้ในตัวแปร Total
8. จบการทำงานของโปรแกรม

6. ฟังก์ชัน (Flow Chart)

