

## บทที่ 6

### โครงสร้างข้อมูลและคำสั่งจัดการข้อมูล

#### 6.1 ตัวเลข คำ และรายการ

บทบาทของคอมพิวเตอร์ คือ การประมวลผลข้อมูลหรือการจัดการข้อมูล (data processing) ดังนั้นสิ่งสำคัญของการสั่งให้คอมพิวเตอร์ทำงานนอกจากคำสั่งต่าง ๆ แล้วก็คือข้อมูล สำหรับภาษาโลโกจะมีการจัดการข้อมูล 3 ชนิด คือ ตัวเลข (numbers) คำ (words) และรายการ (list)

##### 6.1.1 ตัวเลข

ตัวเลข คือ จำนวน ซึ่งแต่ละจะสามารถนำมา บวก ลบ คูณ หาร ตลอดจนเปรียบเทียบ เราสามารถสั่งให้แสดงผลการคำนวณและเปรียบเทียบด้วยคำสั่งแสดงผล เช่น คำสั่ง PRINT ซึ่งมีรูปแบบดังนี้

PRINT จำนวน หรือ PRINT นิพจน์

ผลลัพธ์ที่ได้จะแสดงในส่วนแสดงผลข้อความ

#### กิจกรรมที่ 6.1

##### ตัวเลข

- 1) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

PRINT 145

PRINT 4 + 5

PRINT 36 - 6

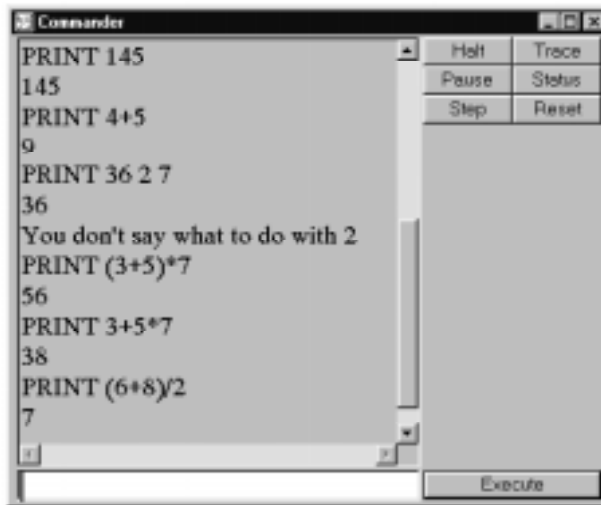
PRINT 36 2 7

PRINT ( 3 + 5 ) \* 7

PRINT 3 + 5 \* 7

PRINT (6+8)/2

2) เปรียบเทียบผลที่ได้กับรูปที่ 6.1



รูปที่ 6.1 การสั่งบวก ลบ คูณ ตัวเลข และผลลัพธ์

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

- บรรทัดที่ 1 สั่งพิมพ์ตัวเลข 145 ผลที่ได้คือ 145
- บรรทัดที่ 2 สั่งบวกจำนวน 4 กับ 5 ผลที่ได้คือ 9
- บรรทัดที่ 3 สั่งลบจำนวน 36 ด้วย 6 ผลที่ได้คือ 30
- บรรทัดที่ 4 เป็นคำสั่งที่ไม่ชัดเจน ผลที่ได้คือ You don't say what to do with 2
- บรรทัดที่ 5 สั่งให้คำนวณนิพจน์  $(3+5)*7$  การใส่วงเล็บที่  $3+5$  จะบอกให้เต่าทำการบวกจำนวน 3 กับ 5 เป็นอันดับแรก จากนั้นจึงนำผลบวกนั้นมาคูณด้วย 7 ผลที่ได้คือ 56
- บรรทัดที่ 6 สั่งให้เต่าคำนวณ  $3+5*7$  กรณีนี้จะทำคูณ 5 ด้วย 7 ก่อนแล้วจึงบวกด้วย 3 นั่นคือเมื่อไม่มีวงเล็บเต่าจะทำการคูณหรือหารก่อนการบวกหรือลบ ผลที่ได้คือ 38
- บรรทัดที่ 7 สั่งให้เต่าคำนวณนิพจน์  $(6+8)*7$  กรณีนี้จะทำการบวก 6 กับ 8 ก่อนแล้วจึงหารด้วย 2 ผลที่ได้คือ 7

3) พิมพ์คำสั่งสำหรับเปรียบเทียบจำนวนในช่องป้อนคำสั่งดังนี้

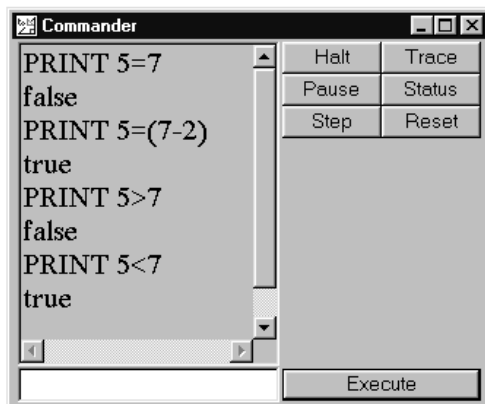
```
PRINT 5 = 7
```

```
PRINT 5 = ( 7 - 2 )
```

```
PRINT 5 > 7
```

```
PRINT 5 < 7
```

4) เปรียบเทียบผลที่ได้กับรูปที่ 6.2



รูปที่ 6.2 การสั่งเปรียบเทียบจำนวน

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

บรรทัดที่ 1 สั่งพิมพ์ผลที่ได้จากการเปรียบเทียบนิพจน์  $5=7$  ซึ่งมีค่าเป็นเท็จ

แต่จึงแสดงผลออกมาเป็น false

บรรทัดที่ 2 สั่งพิมพ์ผลที่ได้จากการเปรียบเทียบนิพจน์  $5=(7-2)$  ซึ่งมีค่าเป็นจริง

แต่จึงแสดงผลออกมาเป็น true

บรรทัดที่ 3 สั่งพิมพ์ผลที่ได้จากการเปรียบเทียบนิพจน์  $5>7$  ซึ่งมีค่าเป็นเท็จ

แต่จึงแสดงผลออกมาเป็น false

บรรทัดที่ 4 สั่งพิมพ์ผลที่ได้จากการเปรียบเทียบนิพจน์  $5<7$  ซึ่งมีค่าเป็นจริง

แต่จึงแสดงผลออกมาเป็น true



จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

บรรทัดที่ 1   สั่งพิมพ์เครื่องหมาย “ โดยไม่มีอะไรตามหลัง เรียกว่า คำที่ว่างเปล่า (empty word) ผลที่ได้คือ บรรทัดว่าง ซึ่งใช้ในการสั่งเว้นบรรทัด

บรรทัดที่ 2-4   สั่งพิมพ์คำต่าง ๆ

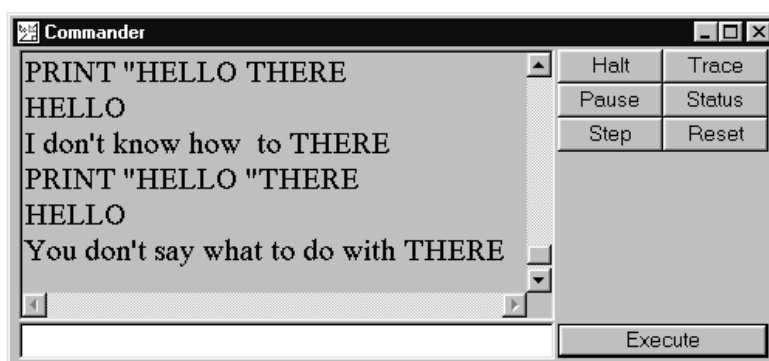
บรรทัดที่ 5   สั่งพิมพ์คำแต่มีเครื่องหมายวงเล็บ แต่ไม่สามารถคาดหมายว่า จะให้ทำอะไร ผลที่ได้คือ unexpected ‘)’

บรรทัดที่ 6-8   สั่งพิมพ์ตัวเลขในลักษณะคำ

3) พิมพ์คำสั่งประมวลผลคำต่อไปในช่องป้อนคำสั่ง

```
PRINT "HELLO THERE
PRINT "HELLO "THERE
```

4) เปรียบเทียบผลที่ได้กับรูปที่ 6.4



รูปที่ 6.4 ผลที่ได้จากตัวอย่างการพิมพ์คำเมื่อมีช่องว่างในประโยค

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

บรรทัดที่ 1   แต่จะถือว่าคำนั้นสิ้นสุดเมื่อพบช่องว่าง ส่วนอื่นที่อยู่หลังช่องว่างจะถือว่า ไม่ใช่ส่วนของคำ จึงพิมพ์ HELLO อย่างเดียว แล้วถือว่า THERE นั้นเป็นกระบวนความ นั่นคือแต่จะบอกว่า I don't know how to THERE

บรรทัดที่ 2   คราวนี้แต่จะรู้ว่า “THERE นั้นเป็นคำ แต่ไม่รู้ว่าจะให้ทำอะไรกับคำนี้ จึงแสดงข้อความ I don't know what to do with THERE

### 6.1.3 รายการ

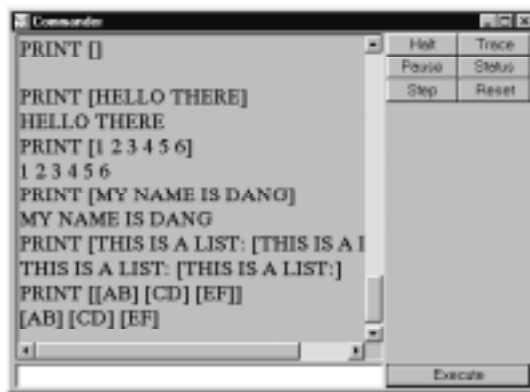
รายการ(list) เป็นการที่จะเชื่อมคำให้เป็นกลุ่มโดยอยู่ภายใต้วงเล็บก้ามปู รายการอาจคลุมทั้ง ตัวเลข คำ และตัวรายการเอง

## กิจกรรมที่ 6.3

### รายการ

- 1) พิมพ์คำสั่งประมวลผลคำต่อไปนี้ในช่องป้อนคำสั่ง
 

```
PRINT []
PRINT [HELLO THERE]
PRINT [1 2 3 4 5 6]
PRINT [MY NAME IS DANG]
PRINT [THIS IS A LIST: [THIS IS A LIST:]]
PRINT [[AB] [CD] [EF]]
```
- 2) เปรียบเทียบผลที่ได้กับรูปที่ 6.5



รูปที่ 6.5 ผลที่ได้จากตัวอย่างการใช้คำสั่งพิมพ์รายการ

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

- บรรทัดที่ 1      สั่งพิมพ์เครื่องหมาย [ ] ซึ่งเรียกว่า รายการว่างเปล่า(empty list)  
ผลที่ได้คือ บรรทัดว่าง ซึ่งใช้ในการพิมพ์เว้นบรรทัด
- บรรทัดที่ 2-4   สั่งพิมพ์รายการต่าง ๆ
- บรรทัดที่ 5      สั่งพิมพ์รายการที่มีเครื่องหมายวงเล็บก้ามปูเป็นส่วนประกอบ
- บรรทัดที่ 6      สั่งพิมพ์รายการที่อยู่ในรายการ

## 6.2 คำสั่งจัดการคำและรายการ

ในการประมวลผลข้อมูล จำเป็นต้องมีคำสั่งจัดการคำและรายการ เพื่อช่วยให้การเขียนโปรแกรมสามารถใช้ข้อมูลต่างชนิดกันได้อย่างมีประสิทธิภาพ คำสั่ง WORD เป็นคำสั่งในการรวมคำหลายคำให้เป็นเป็นคำเดียวกัน และคำสั่ง SENTENCE เป็นคำสั่งที่ใช้ในการรวมคำกับรายการ และ รายการกับรายการ

### 6.2.1 คำสั่ง WORD

คำสั่ง WORD ใช้สำหรับรวมคำ มีรูปแบบดังนี้

WORD “คำสั่ง 1 “คำสั่ง 2

หมายเหตุ ถ้าต้องการเขียนคำมากกว่า 2 คำ ให้ใส่ไว้ในวงเล็บพร้อมกับคำ WORD

## กิจกรรมที่ 6.4

### การใช้คำสั่ง WORD

- 1) พิมพ์คำสั่งประมวลผลคำต่อไปนี้ในช่องป้อนคำสั่ง

PRINT WORD “BIG “WORD

PRINT WORD “WO “RD

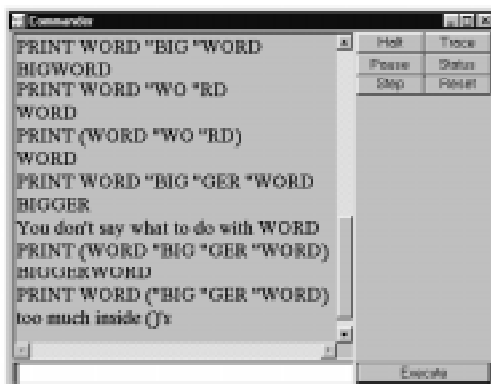
PRINT (WORD “WO “RD)

PRINT WORD “BIG “GER “WORD

PRINT (WORD “BIG “GER “WORD)

PRINT WORD (“BIG “GER “WORD)

- 2) เปรียบเทียบผลที่ได้กับรูปที่ 6.6



รูปที่ 6.6 ผลที่ได้จากตัวอย่างการใช้คำสั่ง WORD

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

- บรรทัดที่ 1   สั่ง PRINT WORD “BIG “WORD   ผลที่ได้คือ BIGWORD  
 บรรทัดที่ 2   สั่ง PRINT WORD “WO “RD   ผลที่ได้คือ WORD  
 บรรทัดที่ 3   คำสั่งและคำอยู่รวมกันในวงเล็บ   ผลที่ได้คือ WORD  
 บรรทัดที่ 4   เมื่อพยายามใช้คำสั่ง PRINT WORD โดยมีคำมากกว่า 2 ตัว ผลที่ได้คือ  
 รวม 2 คำแรกเป็น BIGGER แล้วแจ้งข้อความ You don't say what to do  
 with WORD  
 บรรทัดที่ 5   เมื่อใช้คำสั่ง PRINT WORD โดยมีคำมากกว่า 2 ตัว จะต้องพิมพ์ทั้งคำสั่ง  
 และตัวแปรให้อยู่รวมกันในวงเล็บ   ผลที่ได้คือ BIGGERWORD  
 บรรทัดที่ 6   เมื่อใช้คำสั่ง PRINT WORD ที่มีคำมากกว่า 2 ตัว แต่คำสั่งไม่ถูกจัดรวม  
 อยู่ในวงเล็บร่วมกับคำ   ผลที่ได้คือ to much inside ()'s

## 6.2.2 คำสั่ง SENTENCE

คำสั่ง SENTENCE ใช้ในการเชื่อมคำกับคำ คำกับรายการ และรายการกับรายการ โดยมีรูปแบบดังนี้

- SENTENCE “คำที่ 1 “คำที่ 2   เมื่อต้องการเชื่อมคำกับคำ  
 SENTENCE “คำ [รายการ]   เมื่อต้องการเชื่อมคำกับรายการ  
 หรือ SENTENCE [รายการ] “คำ   เมื่อต้องการเชื่อมคำกับรายการ  
 SENTENCE [รายการ] [รายการ]   เมื่อต้องการเชื่อมรายการกับรายการ

หมายเหตุ 1. ถ้ามีคำหรือรายการมากกว่า 2 ตัวที่จะมาเชื่อมต่อใส่วงเล็บโดยให้นำคำสั่ง

SENTENCE ไว้ในวงเล็บพร้อมกับคำหรือรายการที่ต้องการเชื่อม

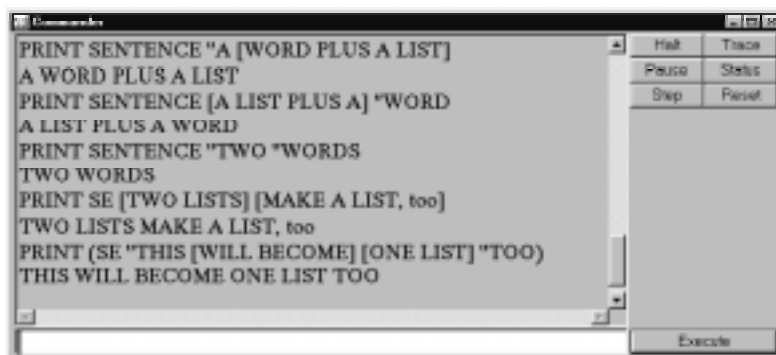
2. SENTENCE อาจใช้ตัวย่อเป็น SE



## กิจกรรมที่ 6.5

### การใช้คำสั่ง SENTENCE

- 1) พิมพ์คำสั่งประมวลผลคำต่อไปนี้ในช่องป้อนคำสั่ง
  - PRINT SENTENCE "A [WORD PLUS A LIST]
  - PRINT SENTENCE [A LIST PLUS A] "WORD
  - PRINT SENTENCE "TWO "WORDS
  - PRINT SE [TWO LISTS] [MAKE A LIST, too]
  - PRINT ( SE " THIS [WILL BECOME] [ONE LIST] "TOO)
- 2) เปรียบเทียบผลที่ได้กับรูปที่ 6.7



รูปที่ 6.7 ผลที่ได้จากตัวอย่างการใช้คำสั่ง SENTENCE

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

บรรทัดที่ 1   สั่ง PRINT SENTENCE "A [WORD PLUS A LIST]

ผลที่ได้คือ A WORD PLUS A LIST

บรรทัดที่ 2   สั่ง PRINT SENTENCE [A LIST PLUS A] "WORD

ผลที่ได้คือ A LIST PLUS A WORD

บรรทัดที่ 3   สั่ง PRINT SENTENCE "TWO "WORDS

ผลที่ได้คือ TWO WORD

บรรทัดที่ 4   สั่ง PRINT SE [TWO LISTS] [MAKE A LIST, too]

ผลที่ได้คือ TWO LISTS MAKE A LIST, too

บรรทัดที่ 5   ใส่วงเล็บ เพื่อเชื่อมคำหรือรายการมากกว่า 2 ตัวโดยสั่ง

PRINT (SENTENCE "THIS [WILL BECOME] [ONE LIST] "TOO)

ผลที่ได้คือ THIS WILL BECOME ONE LIST TOO

### 6.2.3 คำสั่งตัดคำ

โปรแกรม โลโกมีคำสั่งที่ใช้ในการตัดคำ ซึ่งใช้ในการจัดการข้อมูลวิธีหนึ่ง คำสั่งเหล่านี้คือ FIRST LAST BUTFIRST(BF) และ BUTLAST(BL) ใช้เมื่อต้องการส่วนหนึ่งส่วนใดของคำและรายการ

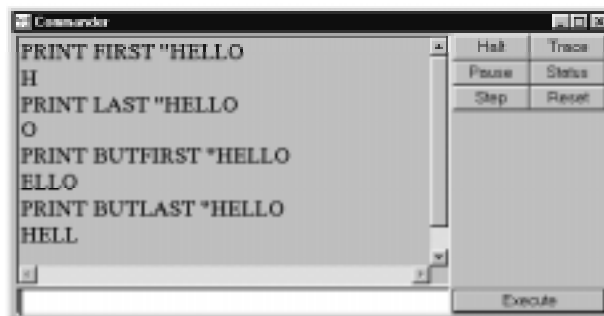
## กิจกรรมที่ 6.6

### การใช้คำสั่งตัดคำ

- 1) สั่งให้เต้าตัดคำที่ต้องการในคำที่กำหนด โดยพิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

```
PRINT FIRST "HELLO
PRINT LAST "HELLO
PRINT BUTFIRST "HELLO
PRINT BUTLAST "HELLO
```

- 2) เปรียบเทียบผลที่ได้กับรูปที่ 6.8



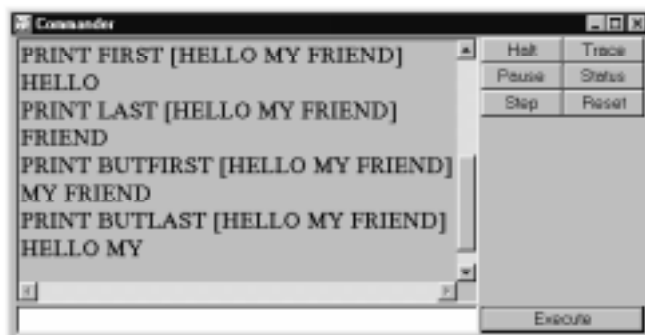
รูปที่ 6.8 ตัวอย่างการใช้คำสั่งตัดคำกับข้อมูล

จากรูป คำสั่ง FIRST และ คำสั่ง LAST จะให้ผลเป็นการตัดคำที่กำหนดให้เหลือเฉพาะตัวแรกหรือตัวสุดท้ายตามลำดับ ส่วนคำสั่ง BUTFIRST และ BUTLAST จะให้ผลเป็นการตัดตัวอักษรตัวแรกหรือตัวสุดท้ายของคำ ตามลำดับ

- 3) พิมพ์คำสั่งต่อไปนี้ในช่องป้อนคำสั่ง

```
PRINT FIRST [HELLO MY FRIEND]
PRINT LAST [HELLO MY FRIEND]
PRINT BUTFIRST [HELLO MY FRIEND]
PRINT BUTLAST [HELLO MY FRIEND]
```

4) เปรียบเทียบผลที่ได้กับรูปที่ 6.9



รูปที่ 6.9 ตัวอย่างการใช้คำสั่งตัดคำกับข้อมูลชนิดรายการ

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

- บรรทัดที่ 1   สั่ง PRINT FIRST [HELLO MY FRIEND]  
ผลที่ได้คือ คำแรกของ รายการได้แก่ HELLO
- บรรทัดที่ 2   สั่ง PRINT LAST [HELLO MY FRIEND]  
ผลที่ได้คือ คำสุดท้ายของรายการได้แก่ FRIEND
- บรรทัดที่ 3   สั่ง PRINT BUTFIRST [HELLO MY FRIEND]  
ผลที่ได้คือ คำทั้งหมดของรายการยกเว้นคำแรกได้แก่ MY FRIEND
- บรรทัดที่ 4   สั่ง PRINT BUTLAST [HELLO MY FRIEND]  
ผลที่ได้คือ คำทั้งหมดของรายการยกเว้นคำสุดท้ายได้แก่ HELLO MY

5) สั่งต่อเชื่อมคำสั่ง FIRST LAST BUTFIRST และ BUTLAST ในช่องป้อนคำสั่ง ดังนี้  
PRINT FIRST BUTFIRST [HELLO MY FRIEND]  
PRINT BUTFIRST FIRST [HELLO MY FRIEND]

6) เปรียบเทียบผลที่ได้กับรูปที่ 6.10



รูปที่ 6.10 ตัวอย่างการเชื่อมคำสั่ง FIRST LAST BUTFIRST และ BUTLAST

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

- บรรทัดที่ 1 พิจารณา BUTFIRST ของ [HELLO MY FRIEND] ก่อน จะได้ MY FRIEND และพิจารณา FIRST ของ MY FRIEND ผลที่ได้คือ MY
- บรรทัดที่ 2 พิจารณา FIRST ของ [HELLO MY FRIEND] ก่อน จะได้ HELLO แล้วพิจารณา BUTFIRST ของ “HELLO ผลที่ได้คือ ELLO

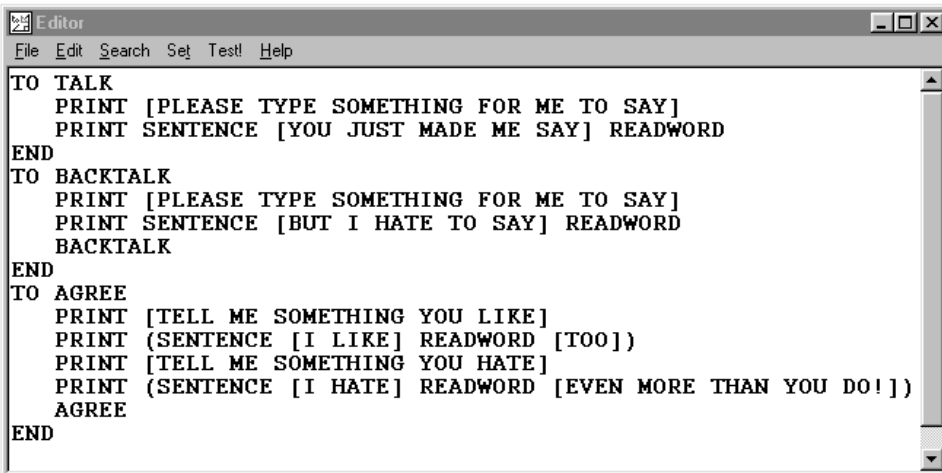
### 6.2.4 คำสั่ง READWORD

การรับข้อมูลในการประมวลผล นอกจากจะใส่ข้อมูลลงในคำสั่งของกระบวนการหรือการเรียกกระบวนการแล้วอาจจะรับข้อมูลโดยการป้อนข้อมูลขณะทำงานได้โดยใช้คำสั่ง READWORD รอผู้ใช้พิมพ์ข้อมูลเข้าไปรอบโต้ตอบ Input Mode ก่อนทำงานตามคำสั่งต่อไป

## กิจกรรมที่ 6.7

### การใช้คำสั่ง READWORD

- 1) สร้างกระบวนการ TALK กระบวนการ BACKTALK และกระบวนการ AGREE ตามลำดับ คำสั่งในรูปที่ 6.11 ดังนี้

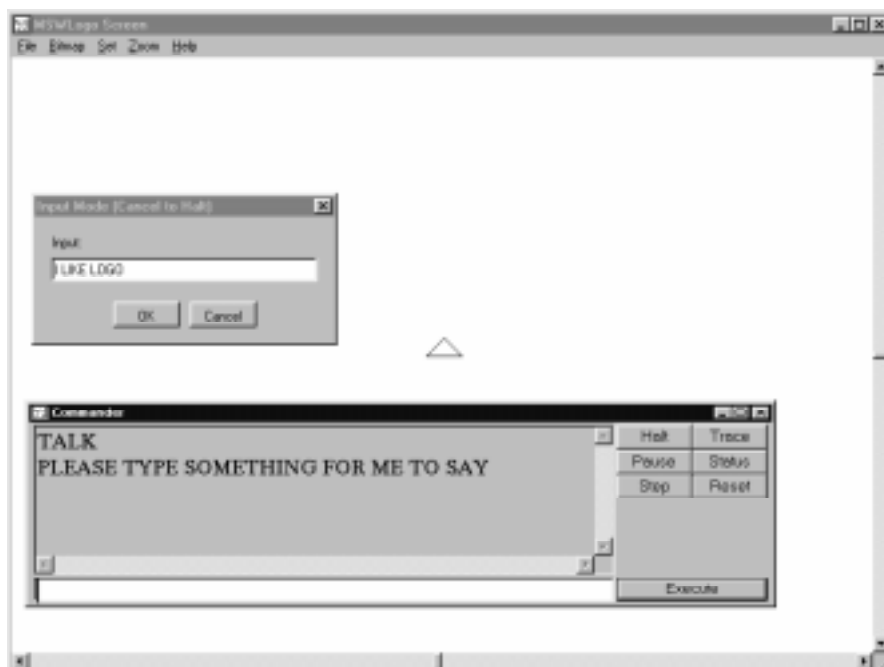


```

Editor
File Edit Search Set Test! Help
TO TALK
  PRINT [PLEASE TYPE SOMETHING FOR ME TO SAY]
  PRINT SENTENCE [YOU JUST MADE ME SAY] READWORD
END
TO BACKTALK
  PRINT [PLEASE TYPE SOMETHING FOR ME TO SAY]
  PRINT SENTENCE [BUT I HATE TO SAY] READWORD
  BACKTALK
END
TO AGREE
  PRINT [TELL ME SOMETHING YOU LIKE]
  PRINT (SENTENCE [I LIKE] READWORD [TOO])
  PRINT [TELL ME SOMETHING YOU HATE]
  PRINT (SENTENCE [I HATE] READWORD [EVEN MORE THAN YOU DO!])
  AGREE
END
  
```

รูปที่ 6.11 ตัวอย่างการใช้คำสั่ง READWORD

- 2) สั่งเต่าให้ทำงานตามคำสั่ง TALK เมื่อปรากฏกรอบโต้ตอบ Input Mode ให้ป้อนข้อความในช่อง Input ว่า I LIKE LOGO ดังรูปที่ 6.12 เมื่อกดแป้น ENTER หรือคลิกปุ่ม OK จะได้ผลดังรูปที่ 6.13



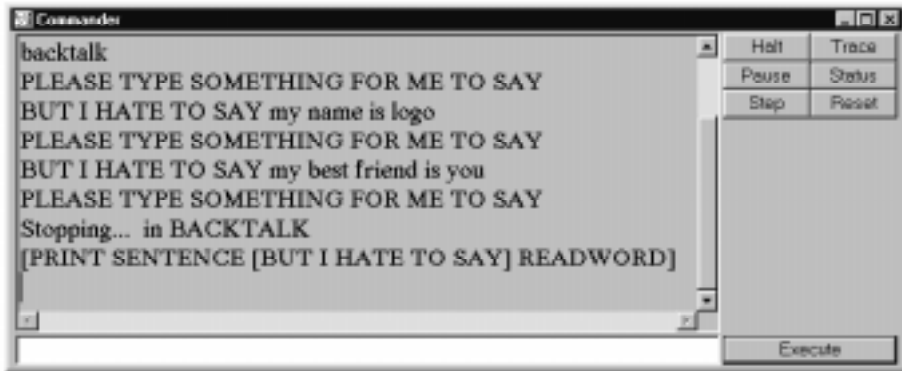
รูปที่ 6.12 แสดงกรอบ Input Mode



รูปที่ 6.13 แสดงผลลัพธ์หลังจากป้อนข้อมูล I LIKE LOGO

จากรูปที่ 6.12 เมื่อเต่าพบคำสั่ง READWORD จะแสดงกรอบโต้ตอบ Input Mode เพื่อให้ป้อนข้อมูล เต่าจะรับข้อมูลนี้ไว้ใน READWORD คำสั่ง SENTENCE จะรวมข้อมูลในรายการกับข้อมูลใน READWORD และถูกพิมพ์ด้วยคำสั่ง PRINT ได้ผลลัพธ์ดังรูปที่ 6.13

- 3) สั่งเต้าให้ทำงานตามคำสั่ง BACKTALK เมื่อปรากฏกรอบโต้ตอบ Input Mode ให้ป้อนข้อความในช่อง Input ว่า my name is logo จะปรากฏผลลัพธ์ว่า I HATE TO SAY my name is logo แล้วจะปรากฏกรอบโต้ตอบ Input Mode อีกครั้งให้ป้อนข้อความในช่อง Input ว่า my best friend is you แล้วกดแป้น ENTER หรือคลิกปุ่ม เมื่อได้ผลลัพธ์แล้วปรากฏกรอบ Input Mode อีก ให้คลิกปุ่ม Cancel จะได้ผลดังรูปที่ 6.14



รูปที่ 6.14 แสดงผลลัพธ์ในการใช้กระบวนการ BACKTALK

- 4) สั่งเต้าให้ทำงานตามคำสั่ง AGREE และให้ป้อนข้อความโต้ตอบกับเต้าตามต้องการ ผลที่ได้คือ .....
- .....
- .....
- .....
- .....
- .....
- .....
- .....

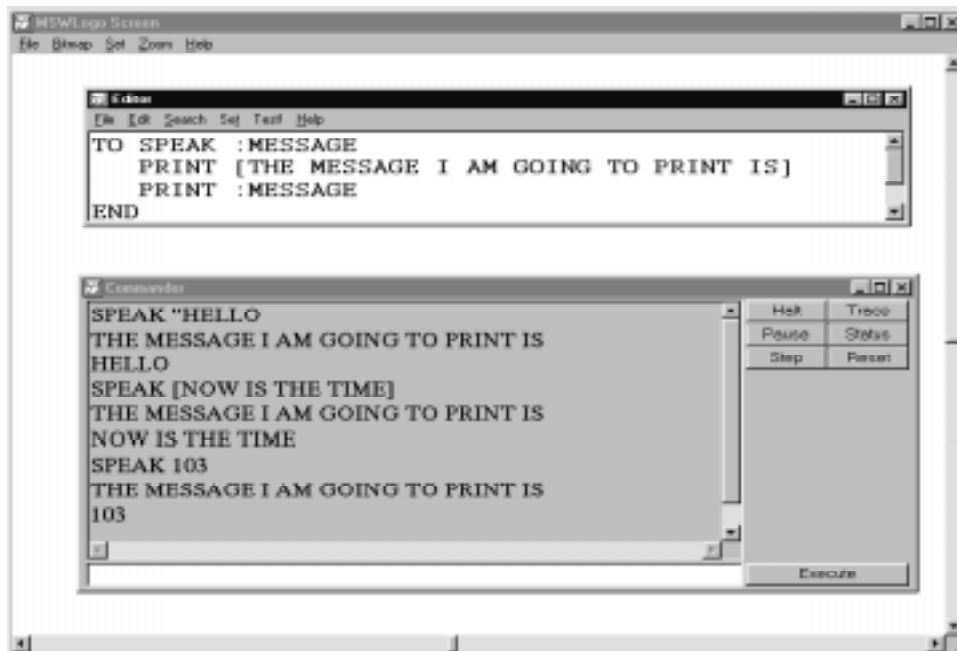
### 6.3 การใช้ค่า และรายการ เป็นค่าตัวแปร

ในบทที่ 5 ได้กล่าวถึงกระบวนการที่มีตัวแปร โดยค่าตัวแปรที่กำหนดให้เป็นข้อมูลตัวเลขใช้สำหรับเปลี่ยนค่าของขนาดหรือรูปร่างของรูปภาพไปแล้ว กิจกรรมในหัวข้อนี้จะกล่าวถึงกระบวนการที่มีตัวแปร โดยค่าของตัวแปรจะเป็นค่าหรือรายการเพื่อสามารถเปลี่ยนแปลงการประมวลค่าได้หลายรูปแบบโดยใช้กระบวนการเดียว

## กิจกรรมที่ 6.8

### การกำหนดค่าตัวแปรเป็น คำ รายการ และตัวเลข

- 1) สร้างกระบวนการความ SPEAK :MESSAGE ตามลำดับคำสั่งในรูปที่ 6.15 ดังนี้



รูปที่ 6.15 ลำดับคำสั่งในกระบวนการความ SPEAK :MESSAGE และผลการใช้คำสั่ง

- 2) สั่งเต่าให้ทำงานด้วยคำสั่งต่อไปนี้

SPEAK "HELLO ผลที่ได้คือ.....

ค่าที่กำหนดให้กับตัวแปร MESSAGE คือ..... เป็นข้อมูลชนิด .....

SPEAK [NOW IS THE TIME] ผลที่ได้คือ.....

ค่าที่กำหนดให้กับตัวแปร MESSAGE คือ..... เป็นข้อมูลชนิด .....

SPEAK 103 ผลที่ได้คือ.....

ค่าที่กำหนดให้กับตัวแปร MESSAGE คือ..... เป็นข้อมูลชนิด .....

- 3) เปรียบเทียบผลที่ได้กับรูปที่ 6.15

## กิจกรรมที่ 6.9

### การใช้คำสั่ง MAKE กำหนดค่าตัวแปรเป็น คำ รายการ และตัวเลข

- 1) สั่งเต้าให้ทำงานตามคำสั่งต่อไปนี้ในช่องป้อนคำสั่ง
 

```
MAKE "MESSAGE [THIS IS GETTING SILLY]
PRINT :MESSAGE
MAKE "MESSAGE "HELLO
PRINT :MESSAGE
MAKE "MESSAGE 1000
PRINT :MESSAGE 1000
PRINT "MESSAGE
PRINT :MESSAGE
```
- 2) เปรียบเทียบผลที่ได้กับรูปที่ 6.16

```

Commander
MAKE "MESSAGE [THIS IS GETTING SILLY]
PRINT :MESSAGE
THIS IS GETTING SILLY
MAKE "MESSAGE "HELLO
PRINT :MESSAGE
HELLO
MAKE "MESSAGE 1000
PRINT :MESSAGE
1000
PRINT "MESSAGE
MESSAGE
PRINT :MESSAGE
1000
  
```

รูปที่ 6.16 ตัวอย่างการใช้คำสั่ง MAKE และผลที่ได้

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

- บรรทัดที่ 1 กำหนดให้ตัวแปร MESSAGE เก็บค่ารายการ THIS IS GETTING SILLY
- บรรทัดที่ 2 พิมพ์ค่าของตัวแปร MESSAGE ผลที่ได้คือ THIS IS GETTING SILLY
- บรรทัดที่ 3 กำหนดให้ตัวแปร MESSAGE เก็บค่าคำ HELLO
- บรรทัดที่ 4 พิมพ์ค่าของตัวแปร MESSAGE ผลที่ได้คือ HELLO
- บรรทัดที่ 5 กำหนดให้ตัวแปร MESSAGE เก็บค่าตัวเลข 1000
- บรรทัดที่ 6 พิมพ์ค่าของตัวแปร MESSAGE ผลที่ได้คือ 1000
- บรรทัดที่ 7 พิมพ์ค่า MESSAGE (ใช้เครื่องหมาย “ นำหน้า) ผลที่ได้คือ MESSAGE
- บรรทัดที่ 8 พิมพ์ค่าของตัวแปร MESSAGE (ใช้เครื่องหมาย : นำหน้า) ผลที่ได้คือ 1000

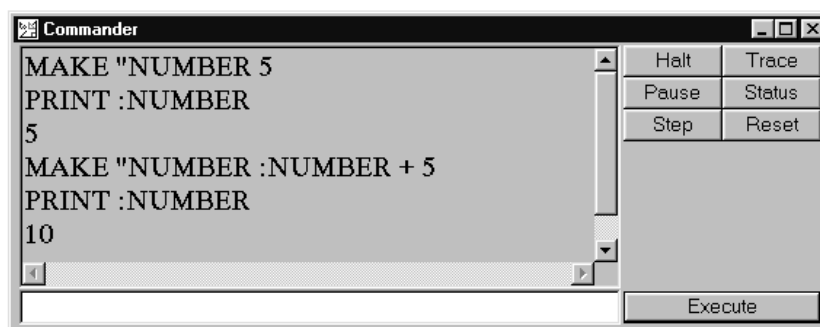


## กิจกรรมที่ 6.10

### การใช้คำสั่ง MAKE เพื่อเปลี่ยนค่าของตัวแปรชนิดตัวเลข

- 1) สั่งเต่าด้วยคำสั่งต่อไปนี้ในช่องป้อนคำสั่ง
 

```
MAKE "NUMBER 5
PRINT :NUMBER
MAKE "NUMBER :NUMBER + 5
PRINT :NUMBER
```
- 2) เปรียบเทียบผลที่ได้กับรูปที่ 6.17



รูปที่ 6.17 ตัวอย่างการใช้คำสั่ง MAKE เพื่อเปลี่ยนค่าของตัวแปรชนิดตัวเลข และผลที่ได้

จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

บรรทัดที่ 1 กำหนดให้ตัวแปร NUMBER เก็บค่าตัวเลข 5

บรรทัดที่ 2 สั่งพิมพ์ ค่าของตัวแปร NUMBER ผลที่ได้คือ 5

บรรทัดที่ 3 กำหนดให้ตัวแปร NUMBER เก็บค่า :NUMBER+5 คือ ให้บวก 5 เข้ากับ  
ค่าของตัวแปร :NUMBER ตัวแปร NUMBER จะเก็บค่าใหม่คือ 10

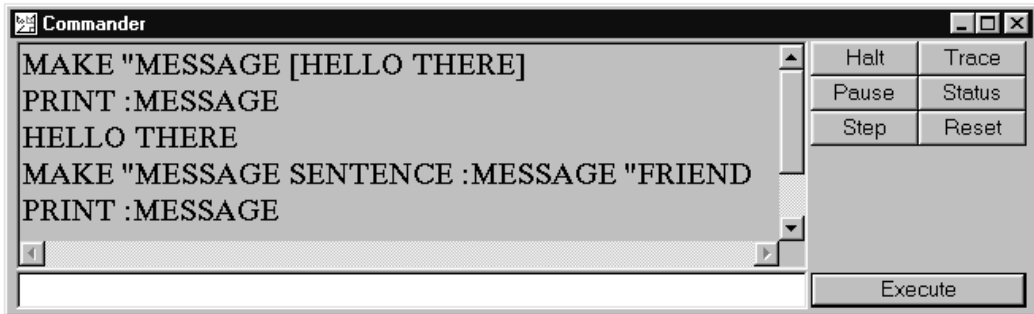
บรรทัดที่ 4 สั่งพิมพ์ ค่าของตัวแปร NUMBER ผลที่ได้คือ 10

## กิจกรรมที่ 6.11

### การใช้คำสั่ง MAKE เพื่อเปลี่ยนค่าของตัวแปร ชนิดรายการ หรือค่า

- 1) สั่งเต่าด้วยคำสั่งต่อไปนี้ในช่องป้อนคำสั่ง
 

```
MAKE "MESSAGE [HELLO THERE]
PRINT :MESSAGE
MAKE "MESSAGE SENTENCE :MESSAGE "FRIEND
PRINT :MESSAGE
```
- 2) เปรียบเทียบผลที่ได้กับรูปที่ 6.18



รูปที่ 6.18 ตัวอย่างการใช้คำสั่ง MAKE เพื่อเปลี่ยนค่าของตัวแปรชนิดรายการหรือค่า  
และผลที่ได้

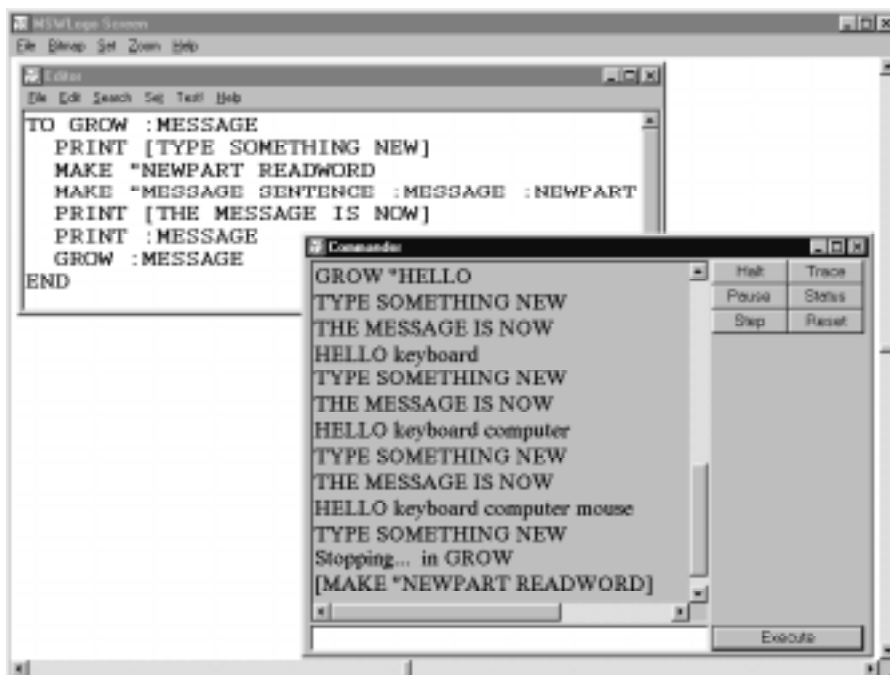
จากรูปแสดงการทำงานของคำสั่งในแต่ละบรรทัดดังนี้

- บรรทัดที่ 1 กำหนดให้ตัวแปร MESSAGE เก็บค่ารายการ HELLO THERE
- บรรทัดที่ 2 สั่งพิมพ์ ค่าของตัวแปร MESSAGE ผลที่ได้คือ HELLO THERE
- บรรทัดที่ 3 กำหนดให้ตัวแปร MESSAGE เก็บค่า :MESSAGE "FRIEND โดยใช้คำสั่ง SENTENCE รวมข้อความที่เก็บในตัวแปร MESSAGE กับ FRIEND ค่าใหม่ของตัวแปร MESSAGE จึงเปลี่ยนเป็น HELLO THERE FRIEND
- บรรทัดที่ 4 สั่งพิมพ์ค่าของตัวแปร MESSAGE ผลที่ได้คือ HELLO THERE FRIEND

## กิจกรรมที่ 6.12

### การกำหนดค่าตัวแปรเป็น คำ รายการ และตัวเลข

- 1) สร้างกระบวนการความ GROW :MESSAGE ตามลำดับคำสั่งในรูปที่ 6.19 ดังนี้



รูปที่ 6.19 ลำดับคำสั่งในกระบวนการความ GROW :MESSAGE และผลการใช้คำสั่ง

- 2) สั่งเต่าให้ทำงานด้วยคำสั่งต่อไปนี้

GROW "HELLO

ถ้าต้องการหยุดป้อนข้อความให้คลิกปุ่ม Cancel ในกรอบโต้ตอบ Input Mode

ผลที่ได้คือ รายการข้อความที่ป้อนเข้าไปทั้งหมด โดยมี HELLO นำหน้า เปรียบเทียบผลที่ได้กับรูปที่ 6.19

## แบบฝึกหัดบทที่ 6

1. จงบอกผลลัพธ์ที่ได้ เมื่อสั่งให้เต้าทำงานตามคำสั่งต่อไปนี้
 

1.1 PRINT 45+23	1.7 PRINT (9-5) <(8-4)
1.2 PRINT -19+(6+23)*2	1.8 PRINT "My name is Sri
1.3 PRINT (48-20)/4	1.9 PRINT [My name is Sri]
1.4 PRINT (22-10)>8	1.10 PRINT [My] [name] [is Sri]
1.5 PRINT (12-7) =5	1.11 PRINT [[My] [name] [is Sri]]
1.6 PRINT :hello	1.12 PRINT [My name] "is Sri
  
2. จงบอกผลลัพธ์ที่ได้ เมื่อสั่งให้เต้าทำงานตามคำสั่งต่อไปนี้ ในช่องป้อนคำสั่ง
  - 2.1 PRINT WORD "My "COMPUTER
  - 2.2 PRINT (WORD "LOGO "TEACH "ME "ABOUT "CONCEPT "OF "PROGRAMMING)
  - 2.3 PRINT SE[LOGO TEACH ME] [ABOUT CONCEPT OF PROGRAMMING]
  - 2.4 PRINT SE[LOGO TEACH] "ME [ABOUT CONCEPT] "OF "PROGRAMMING
  
3. จงบอกผลลัพธ์ที่ได้ เมื่อสั่งให้เต้าทำงานตามลำดับคำสั่งต่อไปนี้ ในช่องป้อนคำสั่ง
 

```
MAKE "MESSAGE [NUMBERS WORD LIST]
PRINT :MESSAGE
PRINT FIRST :MESSAGE
PRINT LAST :MESSAGE
PRINT BF :MESSAGE
PRINT BF LAST :MESSAGE
PRINT BL BF :MESSAGE
```
  
4. จงปรับปรุงกระบวนการความ GROW :MESSAGE ในกิจกรรมที่ 6.11 ให้หยุดทำงานเมื่อป้อนคำว่า Exit