Section G: Built-in Functions and Pseudovariables

ในหัวข้อนี้จะกล่าวถึง built-in ฟังก์ชันทั้งหมดซึ่งโปรแกรมเมอร์จะนำเอาไปใช้ได้ โดยจะเรียงลำดับตามตัวอักษร ส่วน built-in ฟังก์ชัน ซึ่งจะเอาไปใช้เป็นตัวแปรเทียม (pseudo-variable) จะเขียนแยกไว้ต่างหากในหัวข้อเรื่องตัวแปรเทียม

รูปแบบทั่วไปของ built-in function reference มีดังนี้

function name [(x) | (x_1, x_2,....., x_n)]

เมื่อ x หรือ x, x_2,....., x_n หมายถึง arguments ที่ต้องมีสำหรับบางฟังก์ชันอาจจะมี argument เพียงตัวเดียว, บางฟังก์ชันอาจจะมี argument มากกว่าหนึ่งตัว, และบางฟังก์ชันอาจจะไม่มี argument เลยก็ได้

ตัวอย่าง

SUBSTR (X_1, X_2 [, X_3])

แต่ละฟังก์ชันจะกล่าวถึงรูปแบบทั่วไปของ function reference (ส่วน pseudovariable reference จะมีความหมายอย่างเดียวกับ function reference) ซึ่งรายละเอียดของแต่ละฟังก์ชันจะมีหัวข้อต่อไปนี้

1. รายละเอียดของมูลค่าฟังก์ชัน ซึ่งจะส่งกลับ (returned)
2. รายละเอียดของ arguments
3. คุณสมบัติอื่น ๆ ในการใช้ฟังก์ชัน
4. บทประยุกต์, รายละเอียดของความหมายอย่างเดียวกับเมื่อเป็นตัวแปรเทียม

ชนิดของ built-in ฟังก์ชัน

Built-in ฟังก์ชันแบ่งตามลักษณะเฉพาะในภาษา PL/1 ได้เป็น 10 ชนิดดังนี้

String-handling

Arithmetic

Mathematical

Array-handling

Condition-handling

Storage control

Multitasking

Input/Output

Preprocessor

Miscellaneous

4 ชนิดแรก เป็น built-in ฟังก์ชันที่เกี่ยวกับการคำนวณ

1. String-handling built-in functions

ฟังก์ชันเหล่านี้จะกระทำ (process) กับ bit และ character string ได้แก่

BIT	REPEAT
BOOL	STRING
CHAR	SUBSTR
HIGH	TRANSLATE
INDEX	UNSPEC
LENGTH	VERIFY
LOW	

2. Arithmetic built-in functions

ฟังก์ชันกลุ่มนี้มีไว้ให้โปรแกรมเมอร์ ควบคุมการเปลี่ยนรูป base, scale, mode และ precision ทั้งโดยตรงและระหว่างที่มีการทำ operation ทางคณิตศาสตร์ขั้นต้น, ฟังก์ชันอื่น ๆ ในกลุ่มนี้ใช้ตรวจสอบคุณสมบัติอย่างง่ายของมูลค่าคณิตศาสตร์ ตัวอย่างเช่น ฟังก์ชัน SIGN บอกเครื่องหมายของมูลค่าคณิตศาสตร์ ว่าเป็นบวก หรือ ลบ หรือไม่มีเครื่องหมายกำกับ ได้แก่

ABS	IMAG
ADD	MAX
BINARY	MIN

CEIL	MOD
COMPLEX	MULTIPLY
GONIG	PRECISION
DECIMAL	REAL
DIVIDE	ROUND
FIXED	SIGN
FLOAT	TRUNC
FLOOR	

3. Mathematical built-in functions

ฟังก์ชันกลุ่มนี้ให้มูลค่าของ standard mathematical operations ได้แก่

ACOS	LOG
ASIN	LOG 2
ATAN	LOG 10
ATAND	SIN
ATANH	SIND
COS	SINH
COSD	SQRT
COSH	TAN
ERF	TAND
ERFC	TANH
EXP	

4. Array-handling built-in functions

ฟังก์ชันทั้งหมดนี้กระทำกับ array arguments และให้มูลค่าหนึ่งจำนวน ซึ่งเป็นคุณสมบัติของ array นั้น ได้แก่

ALL	LBOUND

ANY	POLY
DIM	PROD
HBOUND	SUM

5. Condition-handling built-in functions

ฟังก์ชันกลุ่มนี้ไว้ให้ โปรแกรมเมอร์ตรวจสอบการ interrupt ซึ่งเกิดจากเงื่อนไขที่รับได้ ฟังก์ชันแต่ละชุดจะให้มูลค่าหนึ่งจำนวน ซึ่งปรากฏเฉพาะในขอบเขตของ an on-unit ซึ่งสามารถนับให้เป็นเงื่อนไขอย่างหนึ่ง ใน built-in ฟังก์ชันหรือภายในขอบเขตของ an on-unit สำหรับเงื่อนไข ERROR หรือ FINISH เมื่อเกิดขึ้นกับ standard system action ได้แก่

DATAFIELD	ONFILE
ONCHAR	ONKEY
ONCODE	ONLOC
ONCOUNT	0NSOURCE

6. Storage control built-in functions

ฟังก์กลุ่มนี้มีไว้ให้โปรแกรมเมอร์ ใช้คำนวณหาเนื้อที่หน่วยความจำที่ใช้และตำแหน่งที่อยู่ของตัวแปร, กำหนดมูลค่าพิเศษให้กับเนื้อที่ และตำแหน่งที่อยู่ของตัวแปร, ทำการเปลี่ยนรูประหว่าง offset และ pointer values, และหาจำนวน generations ของตัวแปรควบคุม ได้แก่

ADDR	NULL
ALLOCATION	OFFSET
EMPTY	POINTER
CURRENTSTORAGE	STORAGE

7. Multitasking built-in functions

ฟังก์ชันกลุ่มนี้มีไว้ให้โปรแกรมเมอร์ ใช้ตรวจสอบสถานะปัจจุบันของ an event variable ได้แก่

COMPLETION

PRIORITY

STATUS

8. Input/output built-in functions

ฟังก์ชันกลุ่มนี้มีไว้ให้โปรแกรมเมอร์ ใช้ตรวจสอบสถานะปัจจุบันของ file หนึ่ง ๆ ได้แก่

COUNT

LINENO

SAMEKEY

9. Preprocessor built-in functions

ฟังก์ชันกลุ่มนี้ เป็น built-jn ฟังก์ชันเฉพาะที่สามารถ execute ได้ด้วย preprocessor ได้แก่

COMPILETIME

COUNTER

INDEX

LENGTH

PARMSET

SUBSTR

ฟังก์ชัน SUBSTR, LENGTH, และ INDEX เป็นฟังก์ชันเดียวกับใน non – preprocessor ซึ่งมีชื่อเหมือนกัน ส่วนฟังก์ชัน COUNTER, COMPILETIME และ PARMSET เป็นฟังก์ชัน ซึ่ง execute ได้เฉพาะโดย preprocessor เท่านั้น

10. Miscellanous built-in functions

ฟังก์ชันพวกนี้ไม่เหมาะสมที่จะจัดให้อยู่ในชุดใด ๆ ที่กล่าวมาแล้ว ได้แก่

DATE

PLIRETV

TIME

ตัวแปรเทียม (Pseudovariables)

built-in ฟังก์ชันบางชุดอาจนำมาใช้เป็นตัวเก็บผลลัพธ์ (receiving fields) ได้ ซึ่งในกรณีนี้เราจะเรียกมันว่าเป็นตัวแปรเทียม และฟังก์ชันนี้จะต้องอยู่ทางซ้ายมือของเครื่องหมายเท่ากับในคำสั่ง assignment หรือ ในคำสั่ง DO, อาจจะอยู่ใน data list ของคำสั่ง GET และอาจจะอยู่ในชื่อ string ของ option KEYTO, STRING, หรือ REPLY แต่ตัวแปรเทียมจะอยู่ซ้อนกันไม่ได้ ตัวอย่างเช่น

```
UNSPEC (SUBSTR (A,1,2)) = '00'B;
```

คำสั่งข้างต้นนี้ใช้ไม่ได้ (invalid)

ตัวแปรเทียมได้แก่

COMPLETION	REAL
COMPLEX	STATUS
IMAG	STRING
ONCHAR	SUBSTR
ONSOURCE	UNSPEC
PRIORITY	

ADD(X_1, X_2, X_3 [,X_4]) Arithmetic

ฟังก์ชันนี้ให้ผลบวกของมูลค่า x_1 กับ x_2 ซึ่งกำหนด precision ด้วย x_3 x_4 และ

เมื่อ x_1 และ x_2 เป็นมูลค่าซึ่งจะนำมาบวกกัน

x_3 เป็นมูลค่าคงที่จำนวนเต็มฐานสิบไม่มีเครื่องหมายกำกับ (unsigned decimal) integer constant) หมายถึงจำนวนเลขทั้งหมดที่ต้องการจากการทำ operation แต่เลขจำนวนนี้ต้องไม่เกินขอบเขตสูงสุดที่กำหนดไว้

x_4 เป็นมูลค่าคงที่จำนวนเต็มฐานสิบ อาจจะมีเครื่องหมายกำกับหรือไม่ก็ได้ หมายถึง scale factor ของผลลัพธ์ สำหรับผลลัพธ์ที่เป็น fixed point ถ้าไม่มี x_4 หมายถึง x_4 เท่ากับศูนย์, สำหรับผลลัพธ์ที่เป็น floating–point ต้องไม่มี x_4

BINARY X_1[, X_2 [,X_3]]) Arithmetic

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลขฐานสอง ซึ่งเป็นมูลค่าของ x_1 ด้วย precision ซึ่งกำหนดโดย x_2 และ x_3

เมื่อ x_1 เป็นมูลค่าซึ่งต้องการให้เปลี่ยนรูปเป็นฐานสอง

x_2 เป็น เลขจำนวนเต็มฐานสิบไม่มีเครื่องหมายกำกับ หมายถึงจำนวนเลขทั้งหมด

ที่ต้องการ จากการทำ operation และเลขจำนวนนี้ ต้องไม่เกินขอบเขตสูงสุดที่กำหนดไว้

x_1 เป็น เลขจำนวนเต็มฐานสิบ อาจจะมีเครื่องหมายกำกับหรือไม่ก็ได้ หมายถึง scale factor ของผลลัพธ์ สำหรับผลลัพธ์ที่เป็น fixed-point ถ้ามี x_2 และไม่มี x_3 นั่นคือ scale factor มีมูลค่าเป็นศูนย์ สำหรับผลลัพธ์ที่เป็น floating-point จะมีเฉพาะ x_2 อย่างเดียว ถ้าไม่มีทั้ง x_2 และ x_3 precision ของผลลัพธ์ ได้จากกฎเกณฑ์การเปลี่ยนรูปของ base

BIT(X_1 [,X_2]) String-handling

ฟังก์ชันนี้ให้ผลลัพธ์เป็น bit string ของมูลค่า x_1

เมื่อ x_1 เป็น expression ซึ่งต้องการให้เปลี่ยนรูป

x_2 เป็น expression ซึ่งจะถูกเปลี่ยนรูปให้เป็นเลขจำนวนเต็ม หมายถึงความยาวของผลลัพธ์ bit string, ถ้าไม่มี x_2 ความยาวนี้ได้จากกฎเกณฑ์ของการเปลี่ยนรูปข้อมูล

CHAR(X_1 [,X_2]) String-handling

ฟังก์ชันนี้ให้ผลลัพธ์เป็น character string ของมูลค่า x_1

เมื่อ x_1 เป็น expression ซึ่งต้องการให้เปลี่ยนรูป

x_2 เป็น expression ซึ่งจะถูกเปลี่ยนรูปให้เป็นเลขจำนวนเต็ม หมายถึง ความยาวของผลลัพธ์ character string, ถ้าไม่มี x_2 ความยาวนี้ได้จากกฎเกณฑ์ของการเปลี่ยนรูปข้อมูล

COMPILETIME Preprocessor

ฟังก์ชันนี้ให้ผลลัพธ์เป็น character string มีความยาว 18 ตัว ประกอบด้วย วัน เดือน ปี และเวลา ที่ใช้แปล source โปรแกรมให้เป็น object โปรแกรม, string ผลลัพธ์จะอยู่ในรูปแบบดังนี้

DD MMM YY HH.MM.SS

เมื่อ หมายถึง Character blank

DD หมายถึง วันที่ของเดือนนั้น

MMM หมายถึง เดือนอยู่ในรูปของ JAN, FEB, MAR, เป็นต้น

YY หมายถึง ปี

HH หมายถึง ชั่วโมง

MM หมายถึง นาที

SS หมายถึง วินาที

เฉพาะเลขศูนย์ซึ่งอยู่ข้างหน้าวันที่ของเดือนเท่านั้นที่จะถูกแทนที่ด้วย character blank ส่วนเลขศูนย์ซึ่งอยู่ข้างหน้า field อื่น ๆ ไม่แทนที่ด้วย blank

ถ้าไม่ได้ใช้เวลาเลย character 8 ตัวสุดท้ายใน string นั้น จะเป็น 00.00.00

COUNTER Preprocessor

ฟังก์ชันนี้ให้ผลลัพธ์เป็น character string ความยาวเท่ากับ 5 ตัว ประกอบด้วยเลขฐานสิบ สำหรับการอ้างถึงครั้งแรกจะให้ผลลัพธ์เป็น 00001 การอ้างครั้งต่อ ๆ ไป ตัวเลขจะเพิ่มขึ้นอีกครั้งละ 1, counter อาจจะใช้สำหรับ generate unique identifiers หรือใช้สำหรับในวัตถุประสงค์ทั่ว ๆ ไป เกี่ยวกับการนับ

ถ้าฟังก์ชัน COUNTER อ้างถึงมากกว่า 99999 ครั้ง มันจะให้ diagnostic และตัวเลข 00000 และถ้ามีการนับครั้งต่อไป มันก็จะถือว่าเป็นการนับครั้งที่ 1

DATE Miscellaneous

ฟังก์ชันนี้ให้ผลลัพธ์เป็น character string ความยาว 6 ตัว อยู่ในรูปแบบดังนี้ yymmdd

เมื่อ yy หมายถึง ปีปัจจุบัน
mm หมายถึง เดือนปัจจุบัน
dd หมายถึง วันที่ปัจจุบัน

DECIMAL(X_1[,X_2[,X_3]]) Arithmetic

ตัวย่อ : DEC (x_1 [,x_2 [,X_3]])

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลขฐานสิบของมูลค่า x_1 ซึ่งกำหนด precision โดย x_2 และ x_3

เมื่อ x_1 เป็นมูลค่าซึ่งต้องการเปลี่ยนรูปให้เป็นเลขฐานสิบ

x_2 เป็นมูลค่าคงที่จำนวนเต็มฐานสิบ ไม่มีเครื่องหมายกำกับ หมายถึงจำนวนเลขทั้งหมดที่ต้องการจากการทำ operation แต่เลขจำนวนนี้ต้องไม่มากกว่าขอบเขตสูงสุดซึ่งกำหนดไว้

x_3 เป็น มูลค่าคงที่จำนวนเต็มฐานสิบ อาจจะมีเครื่องหมายกำกับหรือไม่ก็ได้ หมายถึง scale factor ของผลลัพธ์ สำหรับผลลัพธ์ที่เป็น fixed-point ถ้ากำหนด x_2 แต่ไม่มี x_3 หมายถึง scale factor มีมูลค่าเป็นศูนย์ สำหรับผลที่เป็น floating-point จะมีเฉพาะ x_2 อย่างเดียว

ถ้าไม่มีทั้ง x_2 และ x_3, precision ของผลลัพธ์ ได้จากกฎเกณฑ์การเปลี่ยนรูปของฐานตัวเลข

DIM(X_1,X_2) **Array-handling**

ฟังก์ชันนี้ให้ผลลัพธ์เป็น fixed-point binary integer หมายถึงจำนวนอีลิเม้นต์ทั้งหมด (current extent) ของ array x_1 มี dimension เป็น x_2

เมื่อ x_1 เป็น array ซึ่งต้องมีการจัดสรรเนื้อที่ไว้แล้ว (currently allocated)

x_2 เป็น element expression หมายถึง จำนวนมิติของ x_1 ทั้งนี้ x_1 ต้องมีมิติไม่น้อยกว่า x_2 และต้องไม่เป็น an array of structures

DIVIDE(X_1,X_2,X_3[,X_4]) **Arithmetic**

ฟังก์ชันนี้ให้ผลลัพธ์ผลหารของมูลค่า 2 จำนวน x_1 และ x_2 มี precision เป็น x_3 และ x_4

เมื่อ x_1 เป็น ตัวตั้ง

x_2 เป็น ตัวหาร

x_3 เป็น มูลค่าคงที่ จำนวนเต็มฐานสิบไม่มีเครื่องหมายกำกับ หมายถึงจำนวนเลขทั้งหมด ซึ่งเป็นผลลัพธ์ที่ต้องการจากการทำ operation แต่ตัวเลขนี้ต้องมีมูลค่าไม่มากกว่ามูลคค่าสูงสุดของขอบเขตที่กำหนดไว้

x_4 เป็นมูลค่าคงที่จำนวนเต็มฐานสิบ · อาจจะมีเครื่องหมายกำกับหรือไม่ก็ได้ หมายถึง scale factor ของผลลัพธ์ สำหรับผลลัพธ์ที่เป็น fixed−point ถ้าไม่มี x_4 หมายความว่า scale factor มีมูลค่าเป็นศูนย์ สำหรับผลลัพธ์ที่เป็น floating−point จะมีเฉพาะ x_3 อย่างเดียว

FIXED(X_1[,X_2[,X_3]]) **Arithmetic**

ฟังก์ชันนี้ให้ผลลัพธ์เป็นตัวเลข fixed−point ของมูลค่า x_1 ด้วย precision x_2 และ x_3

เมื่อ x_1 เป็นมูลค่าซึ่งต้องการเปลี่ยนรูปให้เป็นเลข fixed−point

x_2 เป็นมูลค่าคงที่ของจำนวนเต็มฐานสิบไม่มีเครื่องหมายกำกับ หมายถึงจำนวนเลขทั้งหมดของผลลัพธ์ที่ต้องการ

x_3 เป็นมูลค่าคงที่จำนวนเต็มฐานสิบ อาจจะมีเครื่องหมายกำกับหรือไม่ก็ได้ หมายถึง scale factor ของผลลัพธ์ ถ้าไม่มี x_3, scale factor จะมีมูลค่าเป็นศูนย์

ถ้าไม่มีทั้ง x_2 และ x_3 สำหรับผลลัพธ์ที่เป็นเลขฐานสอง จะมี precision (15,0) ส่วนผลลัพธ์เลขฐานสิบจะมี precision (5,0)

FLOAT(X_1[,X_2]) **Arithmetic**

ฟังก์ชันให้ผลลัพธ์ตัวเลข floating−point ของมูลค่า x_1 ด้วย precision x_2

เมื่อ x_1 เป็นมูลค่าซึ่งต้องการเปลี่ยนรูปให้เป็นเลข floating−point

x_2 เป็นมูลค่าคงที่จำนวนเต็มฐานสิบไม่มีเครื่องหมายกำกับ หมายถึง จำนวนเลข

ทั้งหมดของผลลัพธ์ที่ต้องการ ถ้าไม่มี x_2 ผลลัพธ์ซึ่งเป็นเลขฐานสอง จะมี precision (21) ส่วนผลลัพธ์ซึ่งเป็นเลขฐานสิบ จะมี precision (6)

FLOOR(X) Arithmetic

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลขจำนวนเต็ม มูลค่าสูงสุด ซึ่งน้อยกว่าหรือเท่ากับมูลค่า x ในที่นี้ x ต้องเป็น real

ถ้า x เป็นตัวเลข fixed-point มี preeision (p,q) precision ของผลลัพธ์จะถูกกำหนดโดย

(MIN (N, MAX (p − q + 1)),0)

เมื่อ N เป็นตัวเลขสูงสุดของตัวเลขที่จะเป็นไปได้

ตัวอย่าง

```
DCL A FIXED DEC (3,2) INIT (5.74);
DCL (B,C) FIXED DEC (3,2);
    B = CEIL (A); /* ANSWER = 6.00*/
    C = FLOOR (A); /* ANSWER = 5.00*/
```

HBOUND(X_1,X_2) Array-handling

ฟังก์ชันนี้ให้ผลลัพธ์เป็นตัวเลข fixed-point binary integer ซึ่งหมายถึง ขอบเขตสูงสุดปัจจุบัน (current upper bound) ของ array x_1 มี dimension เป็น x_2

เมื่อ x_1 เป็น array ที่กำหนดให้ และต้องมีการจัดสรรเนื้อที่ให้แล้ว

x_2 เป็น an element expression หมายถึง dimension ของ x_1 ทั้งนี้ x_1 ต้องมี dimension ไม่น้อยกว่า x_2 และต้องไม่เป็น an array of structure

INDEX(X_1,X_2) String-handling

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลขจำนวนเต็ม บอกตำแหน่งเริ่มต้นภายใน string x_1 ของ substring ที่เหมือนกับ string x_2

เมื่อ x_1 เป็น string ที่จะทำการตรวจค้น

x_2 เป็น string ที่จะทำการค้นหาจากใน string x_1

ถ้าไม่มี x_2 ใน x_1 เครื่องคอมพิวเตอร์จะให้ผลลัพธ์เป็นเลขศูนย์

ถ้ามี x_2 ใน x_1 มากกว่า 1 ครั้ง เครื่องจะให้ผลลัพธ์เฉพาะตำแหน่งเริ่มต้นที่พบครั้งแรกเท่านั้น

ถ้า argument ตัวใดตัวหนึ่ง เป็น character หรือ เลขฐานสิบจะถูกเปลี่ยนรูปให้เป็น character string, ถ้า argument ตัวใดตัวหนึ่งเป็นเลขฐานสอง หรือทั้งสองตัวเป็นเลขฐานสอง

จะถูกเปลี่ยนรูปให้เป็น bit string

ตัวอย่าง

```
K = INDEX ('POSITION','TI');
K = INDEX ('POSITION','A');
K = INDEX ('POSITION','I');
```

บรรทัดแรก string 'TI' อยู่ใน string 'POSITION' เริ่มต้นที่ตำแหน่งที่ 5 ฉะนั้น K จึงมีมูลค่าเท่ากับ 5

บรรทัดที่ 2 ไม่มี string 'A' ใน string 'POSITION' ฉะนั้น K จึงมีมูลค่าเป็นศูนย์

บรรทัดที่ 3 มี string 'I' อยู่ใน 'POSITION' 2 แห่ง ตำแหน่งแรกที่พบ string 'I' จะเป็นมูลค่าของ K เพราะฉะนั้นในตัวอย่างนี้ K จึงมีมูลค่าเท่ากับ 4

LENGTH(X) String-handling

ฟังก์ชันนี้ให้ผลลัพธ์เป็นตัวเลขจำนวนเต็ม บอกความยาวปัจจุบันของ string x ถ้า x เป็นเลขฐานสองจะถูกเปลี่ยนรูปให้เป็น bit string ถ้าเป็นอย่างอื่น จะถูกเปลี่ยนรูปให้เป็น character string

ตัวอย่าง

```
M = LENGTH ('BANGKOK')
```

string 'BANGKOK' ประกอบด้วยตัวอักษร 7 ตัว ฉะนั้นมูลค่าของ M จึงเท่ากับ 7

LBOUND(X_1,X_2) Array-handling

ฟังก์ชันนี้ให้ผลลัพธ์เป็นตัวเลขจำนวนเต็ม บอกขอบเขตต่ำสุดปัจจุบัน (current lower bound) ของ array x_1 ซึ่งมี dimension เป็น x_2

เมื่อ x_1 เป็น array ที่กำหนดให้ และต้องมีการจัดสรรเนื้อที่ให้แล้ว

x_2 เป็น element expression บอกจำนวนมิติของ x_1

ทั้งนี้ x_1 ต้องมี dimension ไม่น้อยกว่า x_2 และต้องไม่เป็น an array of structure

LINENO(X) Input/Output

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลขจำนวนเต็ม บอกบรรทัดที่ปัจจุบัน (current line number) ของ file x ที่ทำให้

เมื่อ x เป็น file expression, และ file นี้ต้องมี PRINT attribute

MAX(X_1,X_2...,X_n) Arithmetic

ฟังก์ชันนี้ จะต้องมี arguments ตั้งแต่สองตัวขึ้นไป ให้ผลลัพธ์เป็นมูลค่าของ argument ตัวที่มีมูลค่ามากที่สุดในกลุ่ม

เมื่อ $x_1,x_2,...,x_n$ เป็นกลุ่มของมูลค่าซึ่งต้องการหามูลค่าที่มากที่สุด จำนวน arguments ทั้งหมดของฟังก์ชันจะต้องไม่เกิน 64 ตัว และทุกตัวต้องเป็นเลข real

ถ้า arguments เป็นเลข fixed−point มี precisions:

$(p_1, q_1), (p_2, q_2),...,(p_n, q_n)$

Precision ของผลลัพธ์จะถูกกำหนดโดย

(MIN(N, MAX($p_1-q_1,p_2-q_2,...,p_n-q_n$) + MAX ($q_1-q_2,...,q_n$)),

MAX ($q_1,q_2,...,q_n$))

ถ้า arguments เป็นเลข floating−point เนื่องจากจำเป็นต้องมีการเปลี่ยนรูป และมี precision เป็น $p_1,p_2,p_3,..,p_n$ ผลลัพธ์จะมี precision เป็น MAX ($p_1,p_2,p_3,...,p_n$)

MIN($x_1,x_2,..,x_n$) Arithmetic

ฟังก์ชันนี้จะต้องมี arguments ตั้งแต่สองตัวขึ้นไป ให้ผลลัพธ์เป็นมูลค่าของ arguments ตัวที่มีมูลค่าน้อยที่สุดในกลุ่ม

เมื่อ $x_1,x_2,..,x_n$ เป็นกลุ่มของมูลค่าซึ่งต้องการหามูลค่าที่น้อยที่สุด จำนวน arguments ทั้งหมดของฟังก์ชันนี้จะต้องไม่เกิน 64 ตัว และทุกตัวต้องเลข real

ถ้า arguments เป็นเลข fixed−point มี precision:

$(p_1,q_1,),(p_2,q_2),...,(p_n,q_n)$

precision ของผลลัพธ์จะถูกกำหนดโดย

(MIN(N,MAX($p_1-q_1,p_2-q_2...,p_n-q_n$) + MAX ($q_1,q_2,...,q_n$)),

MAX ($q_1,q_2,...,q_n$))

ถ้า arguments เป็นเลข floating−point เนื่องจากจำเป็นต้องมีการเปลี่ยนรูป และมี precision เป็น $p_1,p_2,p_3,..,p_n$ ผลลัพธ์จะมี precision เป็น MAX ($p_1,p_2,p_3,...,p_n$)

ตัวอย่าง

```
DCL (A,B) FIXED DEC (5,2) INIT (3.50,7.20);
DCL (C,D) FIXED DEC (5,2);
    C = MAX (A,B); /* ANSWER = 7.20 */
    D = MIN (A,B); /* ANSWER = 3.50 */
```

MOD(X_1,X_2) Arithmetic

ฟังก์ชันนี้ให้ผลลัพธ์เป็น R ซึ่งไม่เป็นมูลค่าลบ เป็นตัวเลขน้อยที่สุด โดยที่ $(x_1-R)/x_2 = n$ เมื่อ n เป็นเลขจำนวนเต็ม

และ x_1 เป็นมูลค่าที่กำหนดให้ เป็นตัวตั้ง

x_2 เป็นมูลค่าที่กำหนดให้เช่นกันเป็นตัวหาร

R เป็นมูลค่า ไม่เป็นลบ เป็นตัวเลขน้อยที่สุด

ทั้งนี้ x_1 ต้องเป็นเลข real ถ้า x_1 เป็นบวก, R จะเป็นเศษของการหาร x_1 ด้วย x_2

และ x_2 ต้องเป็นเลข real ถ้า x_2 เป็นศูนย์ จะเกิดเงื่อนไข ZERODIVIDE

ถ้า R เป็น floating-point, precision จะเป็นตัวใดตัวหนึ่งของ x_1 และ x_2 ตัวที่มีมูลค่ามากกว่า, ถ้า R เป็น fixed-point, precision จะถูกกำหนดโดย

$$(MIN(N,p_2-q_2 + MAX(q_1,q_2)), MAX(q_1,q_2)$$

เมื่อ (p_1,q_1) และ (p_2,q_2) เป็น precision ของ x_1 และ x_2 ตามลำดับ

ถ้า x_1 และ x_2 เป็น fixed-point ซึ่งมี scale factor ไม่เหมือนกัน อาจจะมีการปัดเศษของมูลค่า R นั่นคือเกิดเงื่อนไข SIZE ขึ้น

ตัวอย่าง

```
R = MOD (1979,4);
R = MOD (1980,4);
R = MOD (1981,4);
R = MOD (1982,4);
R = MOD (1983,4);
R = MOD (1984,4);
```

บรรทัดแรก คำตอบจะเป็น 3 (เศษ 3)

บรรทัดที่สอง คำตอบจะเป็น 0 (เศษ 0)

บรรทัดที่สาม คำตอบจะเป็น 1 (เศษ 1)

บรรทัดที่สี่ คำตอบจะเป็น 2 (เศษ 2)

บรรทัดที่ห้า คำตอบจะเป็น 3 (เศษ 3)

และ บรรทัดสุดท้าย คำตอบจะเป็น 0 (เศษ 0)

MULTIPLY(X_1,X_2,X_3[,x_4]) **Arithmetic**

ฟังก์ชันนี้ให้ผลลัพธ์ ซึ่งเป็นผลคูณของ x_1 กับ x_2 มี precision เป็น x_3 และ x_4

เมื่อ x_1 และ x_2 เป็นมูลค่า

x_3 เป็นมูลค่าคงที่ จำนวนเต็มฐานสิบ ไม่มีเครื่องหมายกำกับ หมายถึงจำนวนเลขซึ่งเป็นผลลัพธ์จากการทำ operation แต่เลขจำนวนนี้ต้องไม่เกินมูลค่าสูงสุดที่กำหนดไว้

x_4 เป็นมูลค่าคงที่จำนวนเต็มฐานสิบ อาจจะมีเครื่องหมายกำกับหรือไม่มีก็ได้หมายถึง scale factor ของผลลัพธ์ สำหรับผลลัพธ์ซึ่งเป็นเลข fixed-point ถ้าไม่มี x_4, scale

factor จะมีมูลค่าเป็นศูนย์ สำหรับผลลัพธ์ ซึ่งเป็นเลข floating–point จะมีเฉพาะ x_3 อย่างเดียว

PROD(X) Array-handling

ฟังก์ชันนี้ให้ผลลัพธ์ซึ่งเป็นผลคูณของ อีลิเม้นต์ ทุกตัวที่อยู่ใน array x

เมื่อ x เป็น array มี อีลิเม้นต์ เป็น integer หรือ floating–point

ถ้าอีลิเม้นต์ของ x เป็น non–integer fixed–point จะถูกเปลี่ยนรูปให้เป็น floating–point

ถ้าอีลิเม้นต์ของ x เป็น string จะถูกเปลี่ยนรูปให้เป็น integer สำหรับผลลัพธ์ซึ่งเป็น fixed–point integer จะมี precision เป็น (N,O)

เมื่อ N เป็นตัวเลขมากที่สุด ที่จะเป็นไปได้ และ x จะต้องไม่เป็น iSUB defined

SIGN(X) Arithmetic

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลข fixed–point binary integer บอกให้รู้ว่า มูลค่าของ x ที่กำหนดให้นั้นเป็นมูลค่าบวก, ศูนย์ หรือ ลบ ผลลัพธ์เป็นดังนี้

มูลค่าของ x	มูลค่าผลลัพธ์
$x > 0$	+ 1
$x = 0$	0
$x < 0$	− 1

แต่ทั้งนี้ x ต้องเป็นเลข real

ตัวอย่าง

```
DCL  I FIXED DEC (5,2) INIT (311),
       J FIXED DEC (5,2) INIT (−25.6),
(K,L)  FIXED BIN (15);
  K = SIGN (I); /* K = + 1 */
  L = SIGN (J); /* L = − 1 */
```

SIN(X) Mathematic

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลข floating–point หมายถึงค่า sine ของ x

เมื่อ x เป็น expression มีหน่วยเป็น radians

เมื่อ x เป็นเลข complex ผลลัพธ์จะเป็น sin(a)* cosh(b) + i* cos(a)* sin(b)

เมื่อ (a + i * b) หมายถึง x

SIND (X)

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลข floating–point หมายถึงค่า sine ของ x

เมื่อ x เป็น an expression มีหน่วยเป็นองศา (degree) และ x ต้องเป็นเลข real

SQRT(X) Mathematic

ฟังก์ชันนี้ให้ผลลัพธ์เป็นเลข floating–point หมายถึงค่า square root ของ x, ถ้า x เป็น real, จะมีมูลค่าน้อยกว่าศูนย์ไม่ได้ ผลลัพธ์จะเป็น positive square root ของ x

SUBSTR(X_1,X_2[,X_3]) String-handling

ฟังก์ชันนี้ให้ผลลัพธ์เป็น substring ของ string x_1

เมื่อ x_1 เป็น string ซึ่งต้องการจะแบ่ง

x_2 เป็นเลขจำนวนเต็มบอกตำแหน่งเริ่มต้นของ substring ใน x_1

x_3 เป็นเลขจำนวนเต็มบอกความยาวของ substring ใน x_1 ถ้า x_3 เป็นศูนย์ จะให้ผลลัพธ์เป็น null string ถ้าไม่มี x_3 substring ผลลัพธ์จะเริ่มตั้งแต่ตำแหน่ง x_2 ใน x_1 จนถึง character ตัวสุดท้ายใน x_1

ถ้า x_1 เป็นเลขฐาน จะถูกเปลี่ยนรูปให้เป็น bit string หรือถ้า x_1 เป็นเลขฐานสิบจะถูกเปลี่ยนรูปให้เป็น character string

ตัวอย่าง

```
DCL Z CHAR (8) INIT ('ABC △ * XBC'), Y CHAR(1);
    K = INDEX (Z,'BC');
    L = INDEX (Z,'△');
    Y = SUBSTR (Z,INDEX (Z,'*') + 1,1);
```

คำตอบ

K = 2,L = 4, Y = 'X'

แบบฝึกหัด

1. จงบอกมูลค่าของฟังก์ชันต่อไปนี้

   INDEX ('PQXYZZXY', 'XYZ')

   SUBSTR ('EDMONTON', 3, 4) || SUBSTR ('PORT CREDIT', 7, 2) ||

   SUBSTR ('CALGARY', 2, 2)

2. จงบอกผลลัพธ์ต่อไปนี้

   a) LENGTH ('ALPHA'||'BET'||'SOUP')

   b) SUBSTR ('HARPO'||'CHICO'||'GROUCHO'||'ZEPPO',11,7)||

      SUBSTR ('HARPO'||'CHICO'||'GROUCHO'||'ZEPPO',6,5)||

      SUBSTR ('HARPO'||'CHICO'||'GROUCHO'||'ZEPPO',18)||

      SUBSTR ('HARPO'||'CHICO'||'GROUCHO'||'ZEPPO',1,5)

   c) SUBSTR ('SUFFIX', LENGTH ('SUFFIX') – 3)

SUM(X) **Array-handling**

ฟังก์ชันนี้ให้ผลลัพธ์เป็นผลบวกของอีลิเม้นต์ ทุกตัวใน array x ที่กำหนดให้

เมื่อ x เป็น an array of arithmetic elements

ถ้าอีลิเม้นต์ของ x เป็นเลข fixed–point ผลลัพธ์จะมี precision เป็น (N,q) เป็น N เป็นจำนวนเลขสูงสุดที่เป็นไปได้, q เป็น scale factor ของ x, ถ้าอีลิเม้นต์ของ x เป็น string ถูกเปลี่ยนรูปให้เป็น integers แต่ทั้งนี้ x ต้องไม่เป็น iSUB–defined

ตัวอย่าง

$$X = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$$

Y = SUM(X);

Z = PROD(X);

คำตอบ Y = 18,Z = 216

TIME Miscellaneous

ฟังก์ชันนี้จะให้ผลลัพธ์เป็น character string ความยาว 9 ตัว อยู่ในรูปแบบ hhmmssttt

เมื่อ hh เป็น ชั่วโมงปัจจุบัน

mm เป็น จำนวนนาที

ss เป็น จำนวนวินาที

ttt เป็น จำนวน milliseconds

ถ้าไม่ได้ใช้เวลาเลย ฟังก์ชันนี้จะให้ผลลัพธ์เป็น (9)'0'

VERIFY(X_1, X_2) String-handling

ฟังก์ชันนี้จะให้ผลลัพธ์เป็นเลข fixed–point binary integer บอกตำแหน่งของ character ตัวแรก หรือ bit แรก ของ string x_1 ซึ่งไม่มีใน string x_2, ถ้า characters ทุกตัว หรือ ทุก bits ใน string x_1 มีใน x_2 ผลลัพธ์จะเป็นศูนย์, ถ้า arguments เป็น arithmetic จะถูกเปลี่ยนรูปให้เป็น string, ถ้า argument ตัวหนึ่งเป็น bit string อีกตัวหนึ่งเป็น character string, ตัวที่เป็น bit string จะถูกเปลี่ยนรูปให้เป็น character string

เมื่อ x_1 เป็น string ซึ่งต้องตรวจว่ามี character ตัวใด ๆ ที่ไม่มีใน x_2

x_2 เป็น verification string ประกอบด้วยกลุ่มของ characters เรียงลำดับอย่างไรก็ได้

ถ้า argument ตัวใดตัวหนึ่งเป็น character หรือเลขฐานสิบ จะถูกเปลี่ยนรูปให้เป็น character string ถ้าตัวใดตัวหนึ่งเป็น bit หรือสองตัวเป็นเลขฐานสอง จะถูกเปลี่ยนรูปให้เป็น bit string

ตัวอย่าง

```
DCL A CHAR (5) INIT ('123.4');
IF VERIFY (A, '0123456789')¬= 0 THEN
                PUT LIST ('NONNUMERIC');
```

เพราะว่า VERIFY (A '0123456789') ให้คำตอบเป็น 4 ดังนั้นเครื่องจะพิมพ์คำว่า NONNUMERIC

แบบฝึกหัด

1. จงศึกษาโปรแกรมข้างล่างนี้

```
BOUND : PROCEDURE OPTIONS (MAIN);
    MAIN : PROCEDURE (M, N, X, Y, Z);
        DECLARE X(*), Y(*), Z(*);
        X(M+1), Y(N+1) = -1E 30; I, J = 1;
        DO K = 1 TO M+N;
          IF X(I) > Y(J) THEN DO; Z(K) = X(I); I = I+1; END;
                         ELSE DO; Z(K) = Y(J); J = J+1; END;
        END;
    END;
    MAJOR : PROCEDURE (M, N);
        DECLARE X(M+1), Y(N+1), Z(M+N);
        GET LIST ((X(K) DO K = 1 TO M), (Y(J)DO J = 1 TO N));
        CALL MAIN (M, N, X, Y, Z);
        PUT PAGE DATA (Z);
    END;
        GET LIST (M, N);
        CALL MAJOR (M, N);
    END:
```

กำหนดให้มีบัตรข้อมูล 3 ใบคือ

บัตรใบที่ 1 12 9

บัตรใบที่ 2 61 57 46 41 30 26 23 21 19 15 12 11

บัตรใบที่ 3 76 66 64 61 59 57 43 32 10

ให้แสดงลักษณะของผลลัพธ์ที่เครื่องจะพิมพ์ออกมาอย่างถูกต้อง

2.
```
G0 : PROC OPTIONS (MAIN);
    DCL QUOTE BIT (1) INIT ('0'B), TEMP CHAR (71),
    (CODE, COMPRESS) CHAR (142) VAR INIT (' '),
    (T1, T2) CHAR (1), SYSIN INPUT, SYSPRINT OUTPUT;
    ON ENDFILE (SYSIN) GO TO L2;
```

```
L1 : DO K = 1 TO 2;
     GET EDIT (TEMP) (COL (2), A(71));
     CODE = CODE || TEMP;
     END L1;
L2 : DO K = 1 TO LENGTH (CODE) - 1;
     T1 = SUBSTR (CODE, K, 1);
     T2 = SUBSTR (CODE, K + 1, 1);
     IF T1 = "" THEN QUOTE = ¬QUOTE;
     IF ¬((T1 = '') & (T2 = '') & (¬QUOTE)) THEN
          COMPRESS = COMPRESS || T1;
     END L2;
     IF T2 ¬= ''THEN COMPRESS = COMPRESS || T2;
     L = LENGTH (COMPRESS);
     NCARDS = (70 + L)/71;
L3 : DO K = 1 TO NCARDS - 1;
     PUT EDIT (SUBSTR(COMPRESS, 71*K - 70, 71) (COL(2),A);
     PUT EDIT (SUBSTR(COMPRESS, 71* (NCARDS - 1) + 1,
          L - 71* (NCARDS - 1)))(COL(2),A);
     END GO:
```

จากโปรแกรมข้างต้นนี้ กำหนดให้มี input data 2 ใบ ดังนี้

คอลัมน์ 2 10 30 72

0012	THIS IS 'PL/1'	PROGRAM
ABCDEF	'2233 44'	VERY EASY TO WRITE

ให้นักศึกษาแสดงผลลัพธ์ที่เครื่องจะพิมพ์ออกมาให้ถูกต้อง

3. จงเขียนโปรแกรมอ่าน character string ความยาวไม่เกิน 200 characters string นี้ให้จบด้วยเครื่องหมาย $ (แต่ไม่นับ $ เป็นส่วนหนึ่งของ string) แล้วเอา blank ทุกตัวที่มีอยู่ใน string นี้ออกให้หมด จากนั้นพิมพ์ string ผลลัพธ์ที่ไม่มี blank โดยไม่ต้องพิมพ์ $

ตัวอย่าง input

'THE FASTER WE FINISH, THE SOONER WE CAN LEAVE$'

พิมพ์

'THEFASTERWEFINISH,THESOONERWECANLEAVE'

4. จงเขียนโปรแกรมอ่านประโยคภาษาอังกฤษหนึ่งประโยค แล้วพิมพ์ประโยคภาษาอังกฤษชุดนั้นย้อนจากคำสุดท้ายมาถึงคำแรก

ตัวอย่าง

I HAVE MY HAIR CUT.

พิมพ์

CUT HAIR MY HAVE I.

5. จงเขียน flowchart และโปรแกรมอ่านมูลค่าของ X(1), X(2),...,X(100) ทั้งหมด 100 ตัว จากบัตรข้อมูล 10 ใบ แล้วคำนวณและพิมพ์มูลค่าของ S^2 จากสูตรต่อไปนี้

$$S^2 = \frac{n(\sum x_i^2) - (\sum x_i)^2}{n(n-1)}$$

เมื่อ n = ขนาดของข้อมูล ในที่นี้มูลค่าเป็น 100

$\sum x_i^2 = x_1^2 + x_2^2 + x_3^2 \ldots + x_{100}^2$

$\sum x_i = x_1 + x_2 + x_3 \ldots + x_{100}$

6. แผนกบริหารงานบุคคลของบริษัทหนึ่ง บันทึกจำนวนชั่วโมงทำงานภายในหนึ่งสัปดาห์ของพนักงานแต่ละคนไว้ในบัตร 80 คอลัมน์หนึ่งใบ โดยมีรายละเอียดดังนี้

คอลัมน์	ข้อมูล	
1-5	รหัสพนักงาน	
7-36	ชื่อ นามสกุล	
38-55	จำนวนชั่วโมงทำงานใน	
	38-40	วันจันทร์
	41-43	วันอังคาร
	44-46	วันพุธ
	47-49	วันพฤหัส
	50-52	วันศุกร์
	53-55	วันเสาร์

ให้นักศึกษาเขียน flowchart และโปรแกรมอ่านบัตรข้อมูลดังกล่าวข้างต้น จำนวนพนักงานไม่เกิน 100 คน แล้วคำนวณว่าพนักงานแต่ละคนใน 1 สัปดาห์นั้น ทำงานทั้งหมดได้

กี่ชั่วโมง และหาค่าเฉลี่ยว่าในหนึ่งวันพนักงานคนนั้นทำงานได้กี่ชั่วโมง กี่นาที จากนั้นให้พิมพ์ผลลัพธ์ดังนี้

คอลัมน์	ข้อมูล
1–5	รหัสพนักงาน
7–36	ชื่อ นามสกุล
41–45	จำนวนชั่วโมงทำงานทั้งหมดใน 1 สัปดาห์
47–52	ค่าเฉลี่ยการทำงานในหนึ่งวัน
	คอลัมน์ 47–49 = จำนวนชั่วโมง
	คอลัมน์ 50–52 = จำนวนนาที

7. บริษัทพลาสติกไทย จำกัด แบ่งหน่วยงานของบริษัทออกเป็น 4 ภาค คือ ภาคเหนือ, ใต้, ตะวันออก และตะวันตกในแต่ละภาคจะมีหน่วยขายอยู่ 5 เขตด้วยกัน ถึงสิ้นปีบริษัทต้องการทราบตัวเลขปริมาณขายทั้งหมด จำแนกตามเขตและภาค ข้อมูลทั้งหมดเจาะในบัตร 80คอลัมน์ ในบัตรแต่ละใบมีรายละเอียดดังนี้

คอลัมน์ 1 หมายเลขเขต (1 = เขต 1, 2 = เขต 2,
3 = เขต 3, 4 = เขต 4, 5 = เขต 5)

คอลัมน์ 2 หมายเลขภาค (1 = ภาคเหนือ, 2 = ภาคใต้
3 = ภาคตะวันออก, 4 = ภาคตะวันตก)

คอลัมน์ 3–12 ปริมาณสินค้าที่ขายมีทศนิยม 2 หลัก

จงเขียน flowchart และโปรแกรมอ่านข้อมูลทั้งหมดจากบัตรจำนวนหนึ่ง แล้วพิมพ์ตารางแจกแจงออกมาดังนี้

เขต	ภาค				รวม
	เหนือ	ใต้	ตะวันออก	ตะวันตก	
1					
2					
3					
4					
5					
รวม					

8. การสร้างตารางค่าเสื่อมราคา โดยใช้วิธีผลรวมของตัวเลขการหาค่าเสื่อมราคา สมมติว่าเรามีทรัพย์สินอยู่ 1 ชิ้น ราคาเมื่อเริ่มต้นเท่ากับ 1,000 บาท ทรัพย์สินชนิดนี้จะเสื่อมราคาภายใน 3 ปี หมายความว่า หลังจาก 3 ปีแล้วมูลค่าจะเป็นศูนย์ ตารางการคำนวณหาค่าเสื่อมราคาคิดดังนี้

ปีที่	อัตราการเสื่อมราคา	ค่าเสื่อมราคา
1	$\frac{3}{1+2+3} = \frac{3}{6}$	$1{,}000 * \frac{3}{6} = 500.00$
2	$\frac{2}{1+2+3} = \frac{2}{6}$	$1{,}000 * \frac{2}{6} = 333.33$
3	$\frac{1}{1+2+3} = \frac{1}{6}$	$1{,}000 * \frac{1}{6} = 166.67$

จะเห็นว่าอัตราการเสื่อมราคาจะเปลี่ยนไปปีต่อปี แต่การคำนวณหาค่าเสื่อมราคายังคงใช้ราคาทรัพย์สินเดิม อัตรานี้จะเป็นตัวเลขเศษส่วน, ตัวเลขส่วนจะเป็นผลรวมของ 1, 2, 3,... ถึงปีที่ทรัพย์สินนี้หมดมูลค่า (ตามตัวอย่าง, ตัวเลขส่วน = 1 + 2 +3 = 6) ตัวเลขเศษจะเป็นปีที่หมดมูลค่าและลดลงไปปีละ 1 ปีต่อปี (ในตัวอย่าง, ตัวเลขเศษเป็น 3, 2, 1 ตามลำดับ)

แต่ถ้าทรัพย์สินนั้นเสื่อมราคาภายใน 5 ปี ตัวหารก็จะเป็น 1 + 2 + 3 + 4 + 5 = 15 และตัวเลขเศษปีแรกจะเป็น 5 ปีต่อไปเป็น 4, 3, 2, 1 ตามลำดับ

ให้นักศึกษาเขียน flowchart และโปรแกรมอ่านบัตรข้อมูลซึ่งมีลักษณะดังนี้

คอลัมน์ 1-5	หมายเลขทรัพย์สิน
คอลัมน์ 6-14	มูลค่าเริ่มต้นของทรัพย์สิน (ทศนิยม 2 หลัก)
คอลัมน์ 15-16	ว่างไว้
คอลัมน์ 17-18	จำนวนปีที่ทรัพย์สินนั้นจะหมดมูลค่า
คอลัมน์ 19-80	ว่างไว้

output ให้พิมพ์ดังนี้ (จากตัวเลขสมมติข้างต้น)

ทรัพย์สินหมายเลข 12345		มูลค่าเริ่มต้น 1,000.00 บาท	
ปีที่	ค่าเสื่อมราคา	ค่าเสื่อมราคาสะสม	มูลค่าทรัพย์สิน
1	500.00	500.00	500.00
2	333.33	833.33	166.67
3	166.67	1,000.00	0.00

9. ข้อมูลเป็น character string อยู่ในบัตร 80 คอลัมน์กี่ใบก็ได้ใน string ชุดนี้เป็นคำในภาษาอังกฤษไม่เกิน 100 คำ แต่ละคำมีเครื่องหมาย blank หนึ่งตัวขั้น เมื่อจบคำสุดท้ายแล้วจะมีอีก 1 blank จงเขียน flowchart และโปรแกรมสร้างตารางแจกแจงความถี่ ซึ่งจะบอกว่าแต่ละคำที่แตกต่างกันนั้น ได้ถูกใช้ซ้ำกันกี่ครั้ง ในตารางนี้ให้เรียงตามลำดับตัวอักษรจาก A ไปถึง Z ด้วย

ตัวอย่าง input

'YOU CAN FOOL SOME OF THE PEOPLE ALL OF THE
TIME AND ALL OF THE PEOPLE SOME OF THE TIME
BUT NOT ALL OF THE PEOPLE ALL OF THE TIME

พิมพ์ผลลัพธ์ดังนี้

NO	WORD	FREQUENCY
1	ALL	4
2	AND	1
3	BUT	1
4	CAN	1
5	FOOL	1
6	NOT	1
7	OF	6
8	PEOPLE	3
9	SOME	2
10	THE	6
11	TIME	3
12	YOU	1

10. โรงเรียนมัธยมแห่งหนึ่งมีนักเรียนในชั้น ม.6 อยู่ 5 ห้อง ในแต่ละห้องมีนักเรียนไม่เกิน 50 คน โปรแกรมเรียนมี 4 สาขา

กำหนดให้เก็บคะแนนสอบไล่ของนักเรียนทุกคนไว้ใน array 3 มิติ ชื่อ MARK ดังนี้

MARK(I,J,K)

เมื่อ I = หมายเลขห้อง, I = 1,2,3,4,5

J = เลขที่ของนักเรียน, J = 1,2,3,4,..., นักเรียนคนสุดท้ายในห้อง

K = หมายเลขโปรแกรม, K = 1,2,3,4

ตัวอย่าง

MARK (3,21,2) = 62

หมายถึงนักเรียนเลขที่ 21 อยู่ห้องที่ 3 เรียนโปรแกรม 2 ได้คะแนนสอบไล่ 62 คะแนน

จงเขียน flowchart และโปรแกรมอ่านบัตรข้อมูล ซึ่งมีไม่เกิน 1,000 ใบ, คำนวณแล้วพิมพ์ผลลัพธ์ดังนี้

a) คะแนนเฉลี่ยของนักเรียนในแต่ละห้องเรียน

b) ในแต่ละโปรแกรมมีนักเรียนกี่คน

c) คะแนนเฉลี่ยของนักเรียนในแต่ละโปรแกรม

d) มีนักเรียนระดับ ม.6 ทั้งหมดกี่คน

e) คะแนนเฉลี่ยของนักเรียนทั้งหมด

ลักษณะบัตรข้อมูล

คอลัมน์ 1-2	หมายเลขห้อง
คอลัมน์ 4-5	เลขที่ของนักเรียน
คอลัมน์ 7-8	หมายเลขโปรแกรม
คอลัมน์ 10-12	คะแนนสอบไล่

11. ชมรมคุ้มครองผู้บริโภคกาแฟกลุ่มหนึ่ง ได้ตั้งข้อสังเกตว่าราคาของกาแฟฝาแดงขนาดบรรจุขวด ๆ ละ 283 กรัม ได้เพิ่มสูงขึ้นมากตลอดระยะเวลาที่ผ่านมา จึงทำการสำรวจราคากาแฟชนิดนี้โดยการสุ่มตัวอย่างร้านค้า แล้วบันทึกราคาของกาแฟทุกเดือนระหว่างปี พ.ศ. 2525, 2526 และ 2527 พบว่าการเปลี่ยนแปลงของราคากาแฟอยู่ในช่วงขวดละ 84 บาท ถึง 155 บาท ดังตารางข้อมูลที่แสดงนี้

ปี	เดือน											
	1	2	3	4	5	6	7	8	9	10	11	12
2525	87	89	89	89	85	84	90	100	90	90	90	100
2526	95	95	95	90	85	100	110	120	115	100	110	97
2527	120	125	130	135	140	145	150	155	155	145	147	140

ตัวเลขทั้งหมดนี้ได้บันทึกลงในบัตรข้อมูล ข้อมูลแต่ละปีใช้บัตร 80 คอลัมน์หนึ่งใบ จงเขียน flowchart และโปรแกรมวิเคราะห์การเปลี่ยนแปลงของราคากาแฟในช่วง 3 ปีข้างต้นดังนี้

a) คำนวณเปอร์เซนต์การเพิ่มขึ้นของราคากาแฟแต่ละเดือนโดยใช้สูตร

$$\% \text{ การเพิ่มจากเดือนที่ } i-1 \text{ ถึงเดือนที่ } i = 100*\left(\frac{\text{ราคากาแฟเดือนที่ } i}{\text{ราคากาแฟเดือนที่ } i-1} - 1\right)$$

b) คำนวณราคาเฉลี่ยของกาแฟในแต่ละปี

c) คำนวณเปอร์เซนต์การเพิ่มขึ้นของราคากาแฟเฉลี่ยแต่ละปีจากปี 2525-2526 และจากปี 2526-2527 โดยใช้สูตรทำนองเดียวกับข้อ a)

d) พิมพ์ตารางข้อมูลที่เป็น input ตารางเปอร์เซนต์การเพิ่มของราคากาแฟแต่ละเดือน (จากเดือนที่ 1-2, เดือนที่ 2-3 ,... เดือนที่ 11-12) ราคาเฉลี่ยของกาแฟในแต่ละปี และเปอร์เซนต์การเพิ่มขึ้นของราคากาแฟเฉลี่ยแต่ละปี

12. จงเขียนโปรแกรมคำนวณค่ามัธยฐาน (median) จากข้อมูลซึ่งเป็น array 1 มิติชื่อ X มีอีลิเม้นต์อยู่ N ตัว (N มีมูลค่าไม่เกิน 1000)

a) เขียน subroutine procedure ชื่อ DATA อ่าน array X จากบัตรข้อมูล ในบัตรแต่ละใบจะมีอีลิเม้นต์ อยู่ 10 ตัว

b) เขียน subroutine procedure ชื่อ SORT เรียงมูลค่า อีลิเม้นต์ ใน array X จากน้อยไปหามาก

c) เขียนฟังกชัน procedure ชื่อ MEDIAN คำนวณค่ามัธยฐานของ array X จากสูตร ถ้า N

เป็นเลขคี่, มัธยฐาน = $X\ \frac{(N+1)}{2}$

ถ้า N เป็นเลขคี่, มัธยฐาน = $\frac{X\left(\frac{N}{2}\right) + X\left(\frac{N}{2}+1\right)}{2}$

d) ในโปรแกรมหลัก เรียก โปรแกรมย่อย DATA, SORT และ MEDIAN ตามลำดับแล้วพิมพ์มูลค่ามัธยฐาน

13. จงเขียนโปรแกรมอ่านข้อมูลซึ่งเป็น array 2 มิติจำนวนสี่ชุดข้างล่างนี้

$$A_{2\times2} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B_{2\times3} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}$$

$$C_{2\times2} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \quad D_{2\times3} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \end{bmatrix}$$

แล้วพิมพ์ matrix T ขนาด 4 โรว์ 5 คอลัมน์ ซึ่งมีมูลค่าดังนี้

$$T_{4\times5} = \begin{bmatrix} A_{11} & A_{12} & B_{11} & B_{12} & B_{13} \\ A_{21} & A_{22} & B_{21} & B_{22} & B_{23} \\ C_{11} & C_{12} & D_{11} & D_{12} & D_{13} \\ C_{21} & C_{22} & D_{21} & D_{22} & D_{23} \end{bmatrix}$$

14. ในการทดสอบวิชา PL/1 ครั้งหนึ่งกำหนดให้ข้อสอบเป็นปรนัยจำนวน 100 ข้อ คำถามแต่ละข้อจะมีคำตอบ 5 ตัวเลือก ซึ่งมีตัวเลือกเดียวเท่านั้นที่ถูกต้อง เฉลยคำตอบที่ถูกต้องทั้งหมดและผลสอบของนักศึกษาทั้งหมด n คน ให้เก็บใน array 1 มิติ และ array 2 มิติ ดังนี้

เฉลยคำตอบ

ข้อ	1	2	3	4	5	6		97	98	99	100
	2	1	5	3	4	4		1	2	2	3

ผลสอบของนักศึกษา

	NAME	ข้อ	1	2	3	4	5	6	.....	97	98	99	100
นักศึกษาคนที่ 1	MISS SOMSRI		2	1	4	3	2	4		1	2	2	3
นักศึกษาคนที่ 2	MR. TONGCHAI		1	5	5	3	2	3		3	2	2	1
นักศึกษาคนที่ 3	MR. PAIROJ		1	1	2	3	4	1		2	2	3	4
:													
นักศึกษาคนที่ n	MISS POONSOOK		4	3	5	3	4	3		1	2	4	3

จงเขียน flowchart และโปรแกรมอ่านข้อมูลข้างต้นแล้วพิมพ์ชื่อของนักศึกษา คะแนนสอบเฉพาะนักศึกษาที่สอบได้ตั้งแต่ 60 คะแนนขึ้นไป

15. จงเขียน flowchart และโปรแกรมคำนวณสถิติการขายรายเดือนของสินค้าตัวอย่าง โดยให้ input record มีข้อมูลดังนี้ ชื่อร้านค้า, ที่ตั้งและยอดขายรายเดือน สำหรับ output record แต่ละบรรทัดจะเป็นข้อมูลของร้านค้าแต่ละร้าน จากนั้นพิมพ์บรรทัดผลรวมยอดขายทั้งหมดของสินค้าตัวอย่าง, ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของการขาย

สถิติการขายรายเดือนของสินค้าตัวอย่าง คำนวณได้จากสูตร

1) Sample total $= S_1 + S_2 + S_3 + ... + S_n$

2) Average $= \dfrac{\text{Sample total}}{n}$

3) Standard deviation $= \sqrt{\dfrac{n(S_1^2 + S_2^2 + ... + S_n^2) - (S_1 + S_2 + ... + S_n)^2}{n(n-1)}}$

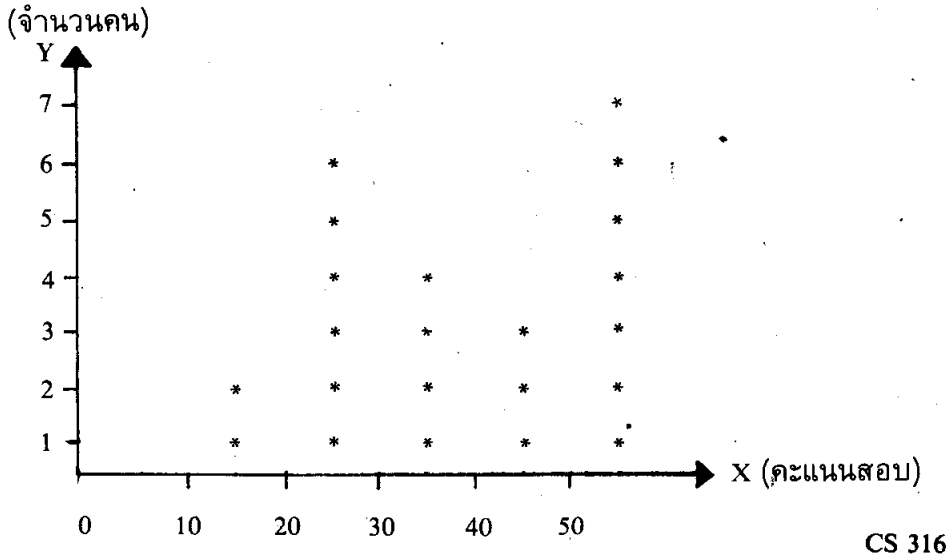
เมื่อ n หมายถึง จำนวนร้านค้าตัวอย่าง

S_i หมายถึง ยอดขายสินค้าตัวอย่างของร้านค้าที่ i

16. ข้อมูลสถิติเกี่ยวกับผลสอบวิชา CS 316 ในภาคฤดูร้อน ปีการศึกษา 2529 มีดังนี้

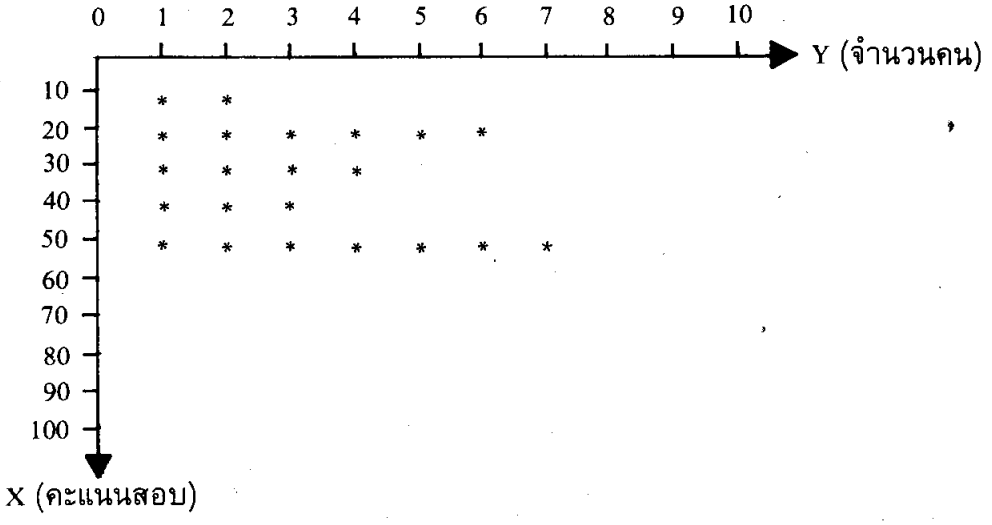
คะแนนระหว่าง	จำนวนคน
0 – 9	0
10 – 19	2
20 – 29	6
30 – 39	4
40 – 49	3
50 ขึ้นไป	7

ข้อมูลชุดนี้ถ้านำมาเขียนเป็นฮิสโตแกรม (Histogram) จะมีลักษณะดังนี้



จงเขียน flowchart และโปรแกรมอ่านข้อมูลซึ่งเป็นคะแนนสอบวิชา CS 316 ของภาค 1 ปีการศึกษา 2530 จาก 0 ถึง 100 คะแนนช่วงละ 10 คะแนนเหมือนกับตัวอย่างข้างต้น จำนวนนักศึกษาที่เข้าสอบมีไม่เกิน 50 คน ในแต่ละช่วงคะแนนมีนักศึกษาไม่เกิน 10 คน แล้วพิมพ์ฮิสโตแกรมของข้อมูลชุดนี้

หมายเหตุ เพื่อให้โปรแกรมเขียนง่ายขึ้น นักศึกษาอาจจะกำหนดให้เส้นแนวนอนเป็นแกน Y และให้เส้นแนวตั้งเป็นแกน X ได้ดังนี้



17. กำหนดเมตริกซ์ A ขนาด n โรว์ m คอลัมน์ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

A′ (transpose ของ A) หมายถึง

$$\begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & & & \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{bmatrix}$$

นั่นคือ transpose ของเมตริกซ์ A ได้จากการสลับที่ระหว่างโรว์กับคอลัมน์

a) จงเขียน function procedure รับเมตริกซ์ A ขนาด n โรว์ m คอลัมน์ แล้วส่งค่า transpose ของเมตริกซ์ A กลับไปยังโปรแกรมเรียก

และเมตริกซ์ A ชุดดังกล่าวจะเป็น symmetric matrix ถ้า $n = m$ และ $a_{ij} = a_{ji}$ ทุกค่า $1 \leqslant i \leqslant n$ และ $1 \leqslant j \leqslant m$

ตัวอย่าง

$$\begin{bmatrix} 1 & 4 & 7 \\ 4 & 2 & 9 \\ 7 & 9 & 3 \end{bmatrix}$$ เป็น symmetric matrix

b) จงเขียน subroutine procedure ตรวจสอบว่า เมตริกซ์ A ซึ่งเป็น input นี้เป็น symmetric matrix หรือไม่

c) กำหนดเมตริกซ์ A และ B ดังนี้

$$A_{nxm} = \begin{bmatrix} a_{11}\ a_{12} \ldots a_{1m} \\ a_{21}\ a_{22} \ldots a_{2m} \\ \vdots \\ a_{n1}\ a_{n2} \ldots a_{nm} \end{bmatrix} \quad \text{และ} \quad B_{mxr} = \begin{bmatrix} b_{11}\ b_{12} \ldots b_{1r} \\ b_{21}\ b_{22} \ldots b_{2r} \\ \vdots \\ b_{m1}\ b_{m2} \ldots b_{mr} \end{bmatrix}$$

ให้เมตริกซ์ C เป็นผลคูณของเมตริกซ์ A และ B

$$\text{ดังนั้น} \quad C_{nxr} = \begin{bmatrix} c_{11}\ c_{12} \ldots c_{1r} \\ c_{21}\ c_{22} \ldots c_{2r} \\ \vdots \\ c_{n1}\ c_{n2} \ldots c_{nr} \end{bmatrix}$$

$$\text{นั่นคือ} \quad c_{ij} = \sum_{k=1}^{m} a_{ik} * b_{kj}$$

ตัวอย่าง

$$A_{2\times3} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad B_{3\times2} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\text{จะได้} \quad C_{2\times2} = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}$$

จงเขียนโปรแกรมอ่านค่าของเมตริกซ์ A และ B ดังกล่าวข้างต้น แล้วคำนวณและพิมพ์เมตริกซ์ C

18. ห้างสรรพสินค้า HOME1 และ HOME2 ได้เจาะชื่อลูกค้าที่เป็นสมาชิกของแผนกเบ็ดเตล็ดไว้ในบัตร 80 คอลัมน์ 2 ใบ ชื่อแต่ละชื่อจะคั่นด้วยเครื่องหมาย comma 1 ตัว สมมติว่าในบัตรแต่ละใบจะไม่มีชื่อซ้ำกันเลย

ตัวอย่าง

สมาชิกของ HOME1 ⌈WONPEN, SAKDA, OPAS, KOVIT, PANEE, SOMSRI, DANG

สมาชิกของ HOME2 ⌈VICHAI, DANG, SOPA, SOMSRI, PANEE, WONPEN

จงเขียนโปรแกรมอ่านบัตรข้อมูล 2 ใบนี้ แล้ว

a) พิมพ์รายชื่อลูกค้าทั้งหมดที่เป็นสมาชิก HOME1 และ/หรือ HOME2 ถ้ามีชื่อซ้ำกันให้พิมพ์ครั้งเดียว (union operation)

b) พิมพ์รายชื่อลูกค้าทั้งหมดที่เป็นสมาชิกทั้ง HOME1 และ HOME2 (intersection operation)

19. จงเขียนโปรแกรมอ่าน input หนึ่งบรรทัดจากเทอร์มินัล โดยสมมติว่า input แต่ละชุดประกอบด้วย 2 ส่วนมีเครื่องหมาย : (colon) หนึ่งตัวคั่นจากนั้นให้พิมพ์ผลลัพธ์เป็นตัวอักขระ 1 ตัว จากกฎเกณฑ์ข้างล่างนี้

N ทั้งบรรทัดไม่มีเครื่องหมาย :

L ส่วนซ้ายมือ (ก่อนเครื่องหมาย :) ยาวกว่าส่วนขวามือ

R ส่วนขวามือ (หลังเครื่องหมาย :) ยาวกว่าส่วนซ้ายมือ

D ทั้งส่วนซ้ายมือและส่วนขวามือมีความยาวเท่ากัน แต่ไม่จำเป็นต้องเหมือนกัน

S ทั้งส่วนซ้ายมือและส่วนขวามือมีความยาวเท่ากันและเหมือนกัน

ตัวอย่าง

Input	**Output**
Sample Sample	N
Short : Long	L
Sample : Sample	S

20. กำหนด array 1 มิติให้ 2 ชุดชื่อ LIST–A มีจำนวนอีลิเม้นต์ n ตัว และ LIST–B มีจำนวนอีลิเม้นต์ m ตัว, อีลิเม้นต์แต่ละตัวมีมูลค่าเป็นชื่อของลูกค้า ข้อมูล 2 ชุดนี้เรียงลำดับชื่อแล้วจากน้อยไปหามาก (A → Z) จงเขียน subroutine procedure สร้าง array 1 มิติอีกชุดหนึ่งให้ชื่อ NEW–LIST ซึ่งเกิดจากการรวม LIST–A และ LIST–B เข้าด้วยกัน ในกรณีที่มีชื่อซ้ำกันให้เก็บเพียงชื่อเดียว ทั้งนี้ array ชุดใหม่นี้มูลค่าของอีลีเม้นต์ทุกตัว ต้องเรียงลำดับจากน้อยไปหามากเช่นกัน

เฉลยข้อ 4

```
 1      0    MAIN : PROCEDURE OPTIONS(MAIN);
 2    1 0          DCL  SYSIN INPUT, SYSPRINT OUTPUT;
 3    1 0          DCL  (I,J,N,M)  FIXED  DEC(4);
 4    1 0          DCL (B,K)  FIXED  DEC(4);
 5    1 0          DCL  STRING     CHAR(80)  VAR INIT('');
 6    1 0          DCL  ST(10)     CHAR(9)  VAR  INIT('');
 7    1 0          DCL  SUBSTR  BUILTIN, INDEX  BUILTIN;
 8    1 0          ON  ENDFILE(SYSIN)  STOP;
 9    1 0              GET  SKIP  EDIT(STRING)  (COL(1),A(80));
10    1 0          PUT  SKIP(20)  EDIT(STRING) (COL(50),A);
11    1 0               I =1;
12    1 0               J = 1;
13    1 0                 DO WHILE (INDEX(ST(I),'.') = 0);
14    1 1                    I = I+1;
15    1 1                    M = INDEX(STRING,' ') - 1;
16    1 1                    N = M+2;
17    1 1                    ST(I) = SUBSTR(STRING,J,M);
18    1 1                    K = I;
19    1 1                    STRING = SUBSTR(STRING,N);
20    1 1                 END;
21    1 0          B = INDEX(ST(I),'.') -1;
22    1 0          ST(I) = SUBSTR(ST(I),1,B);
23    1 0          PUT  SKIP(5)  EDIT((ST(I)  DO I=K TO 2 BY -1),'.')
                                  (COL(50),(K)(X(1),A(6)),A);
24    1 0     END MAIN;
```

```
                    TOYOTA MOTO THAILAND CO,LTD.

                    CO,LTD THAILA MCTO  TOYOTA .
```

เฉลยข้อ 9

```
 1       0     PICT : PROC OPTIONS(MAIN);
 2    1  0            DCL SYSIN INPUT, SYSPRINT OUTPUT, CLS CHAR (10),
                          INDEX BUILTIN, SUBSTR BUILTIN, SENT CHAR (200),
                          R(100) CHAR (10), (I,J,K,L,M) FIXED DEC (3);
 3    1  0            ON ENDFILE(SYSIN) GO TO RINP;
 4    1  0            L, M = 1; I = 0;
 6    1  0            DO WHILE (1 = 1);
 7    1  1            GET FILE(SYSIN) DATA (SENT) COPY;
 8    1  1         A : DO WHILE (1 = 1);
 9    1  2               K = INDEX(SENT,' ');
10    1  2               IF K = 1 THEN LEAVE A;
11    1  2               I = I + 1;
12    1  2               R(I) = SUBSTR(SENT,1,K-1);
13    1  2               SENT = SUBSTR(SENT,K+1);
14    1  2            END;
15    1  1            END;
16    1  0     RINP : DO J = 1 TO I;
17    1  1               DO K = J+1 TO I;
18    1  2                  IF R(J) > R(K) THEN DO; CLS = R(J); R(J) = R(K);
21    1  3                                          R(K) = CLS; END;
23    1  2               END;
24    1  1            END;
25    1  0            PUT SKIP(3) EDIT ('NO', 'WORD', 'FREQUENCY')
                                      (COL(30), A, X(7), A, X(5), A);
26    1  0            DO J = 1 TO I;
27    1  1               IF R(J) = R(J+1) THEN M = M + 1;
28    1  1                                ELSE CALL KUNY;
29    1  1            END;
30    1  0     KUNY : PROCEDURE;
31    2  0            PUT SKIP(2) EDIT (L, R(J), M) (COL(29), F(3), X(7), A, F(4));
32    2  0            M = 1; L = L + 1;
34    2  0            END KUNY;
35    1  0            END PICT;
```

```
SENT='YOU CAN FOOL SOME OF THE PEOPLE ALL OF THE TIME AND ALL OF THE PEOPLE SOME
 OF THE TIME BUT NOT ALL OF THE PEOPLE ALL OF THE TIME ';
```

NO	WORD	FREQUENCY
1	ALL	4
2	AND	1
3	BUT	1
4	CAN	1
5	FOOL	1
6	NOT	1
7	OF	6
8	PEOPLE	3
9	SOME	2
10	THE	6
11	TIME	3
12	YOU	1

เฉลยข้อ 10

```
I OPTIMIZING COMPILER          CAL: PROC OPTIONS (MAIN);

                    SOURCE LISTING

STMT LEV NT

   1      0  CAL: PROC OPTIONS (MAIN);
   2    1 0        DCL (NR(5),NP(4),MARK(5,50,4)) FIXED DEC;
   3    1 0        DCL (SR(5),SP(4),AR(5),AP(4)) FIXED DEC(8,3);
   4    1 0        DCL (I,J,K,N) FIXED BIN;
   5    1 0        DCL SUM BUILTIN;
   6    1 0        DCL SYSIN INPUT,  SYSPRINT OUTPUT;
   7    1 0        ON ENDFILE (SYSIN) BEGIN;
   8    2 0        AR(*)=SR(*)/NR(*);
   9    2 0        AP(*)=SP(*)/NP(*);
  10    2 0        PUT EDIT (('AVERAGE SCORE OF STUD NT IN ROOM',I,'  =',AR(I)
                        DO I=1 TO 5),('# STUDENT WHICH STUEY PROG',I,'  =',NP(I),
                        'AVERAGE SCORE  =',AP(I) DO I=1 TO 4),'TOTAL  =',N  ,
                        'AVERAGE SCORE  =',SUM(SP)/N)  (LINE(15),5 (SKIP(2),X(35),A,
                        F(2),A,F(10,2)),SKIP(2),4 (SKIP(2),X(25),A,F(2),A,F(6),
                        X(10),A,F(10,2)),SKIP(4),X(48),A,F(6),X(10),A,F(10,2));
  11    2 0        STOP;
  12    2 0        END;
  13    1 0        PUT EDIT('ROOM','NO.','PROG','SCORE') (COL(43),2 (A(11),A(9)));
  14    1 0        NR,  SR  =0;          NP,  SP  =0;
  16    1 0        DO N=0 BY 1;
  17    1 1        GET EDIT (I,J,K,MARK(I,J,K)) (3 F(1),F(2));
  18    1 1        PUT EDIT (I,J,K,MARK(I,J,K))  (COL(35),4 F(10));
  19    1 1        NR(I)=NR(I)+1;
  20    1 1        NP(K)=NP(K)+1;
  21    1 1        SR(I)=SR(I)+MARK(I,J,K);
  22    1 1        SP(K)=SP(K)+MARK(I,J,K);
  23    1 1        END;
  24    1 0        END CAL;
```

ROOM	NO.	PROG	SCORE
1	1	1	75
1	2	1	58
1	3	4	45
1	4	3	70
1	5	1	76
1	6	2	90
1	7	2	85
1	8	4	60
2	1	4	65
2	2	3	35
2	3	1	95
2	4	1	80
2	5	4	50
2	6	3	72
3	1	4	82
3	2	2	72
3	3	4	95
3	4	1	75
3	5	3	59
3	6	3	50
4	1	2	67
4	2	1	98
4	3	4	75
4	4	4	50
4	5	2	80
5	1	3	60
5	2	3	76
5	3	1	70
5	4	4	45
5	5	2	85
5	6	1	95
5	7	2	58

AVERAGE SCORE OF STUDENT IN ROOM 1	=	69.88
AVERAGE SCORE OF STUDENT IN ROOM 2	=	66.17
AVERAGE SCORE OF STUDENT IN ROOM 3	=	72.17
AVERAGE SCORE OF STUDENT IN ROOM 4	=	74.00
AVERAGE SCORE OF STUDENT IN ROOM 5	=	69.86

# STUDENT WHICH STUDY PROG 1	=	9	AVERAGE SCORE	=	80.22
# STUDENT WHICH STUDY PROG 2	=	7	AVERAGE SCORE	=	76.71
# STUDENT WHICH STUDY PROG 3	=	7	AVERAGE SCORE	=	60.29
# STUDENT WHICH STUDY PROG 4	=	9	AVERAGE SCORE	=	63.00
TOTAL	=	32	AVERAGE SCORE	=	70.25

เฉลยข้อ 18

```
/*... PROGRAM TO PRINT MENBER OF HOME1 AND HOME2 ...*/

A12 : PROCEDURE OPTIONS (MAIN) ;
      DCL (N1(20),N2(20)) CHAR(15) VAR STATIC ;
      DCL (UN(40),INTER(20)) CHAR(15) VAR ;
      DCL (A,C,I,J,K) FIXED BIN STATIC INIT(0) ;
      DCL (M,N) FIXED BIN STATIC INIT(1) ;
      DCL (IN121,IN122,OUT12) FILE ;
      OPEN FILE (IN121) STREAM INPUT ;
      OPEN FILE (IN122) STREAM INPUT ;
      OPEN FILE (OUT12) STREAM PRINT ;
      ON ENDFILE (IN121) I = 1 ;
      ON ENDFILE (IN122) J = 1 ;
      GET FILE (IN121) LIST (N1(N)) ;
      GET FILE (IN122) LIST (N2(M)) ;
      DO WHILE (I ^= 1) ;
       N = N + 1 ;
       GET FILE (IN121) LIST (N1(N)) ;
      END ;
      DO WHILE (J ^= 1) ;
       M = M + 1 ;
       GET FILE (IN122) LIST (N2(M)) ;
      END ;
      DO I = 1 TO N-1 ;
       K = K + 1 ;
       UN(K) = N1(I) ;
      END ;
      DO J = 1 TO M-1 ;
       DO I = 1 TO N-1 ;
        IF N1(I) = N2(J) THEN DO ;
                               A = A + 1 ;
                               INTER(A) = N1(I) ;
                              END ;
                         ELSE C = C + 1 ;
       END ;
       DO WHILE(C = N-1) ;
        K = K + 1 ;
        UN(K) = N2(J) ;
        C = 0 ;
       END ;
       C = 0 ;
      END ;
```

```
   PUT FILE (OUT12) SKIP EDIT('*** MEMBER OF HOME 1 ***')
            (COL(10),A) ;
   PUT FILE (OUT12) SKIP(2) EDIT(' ')(COL(10),A) ;
   PUT FILE (OUT12) EDIT((N1(I),',' DO I = 1 TO N-2))(2 A) ;
   PUT FILE (OUT12) EDIT(N1(I))(A) ;
   PUT FILE (OUT12) SKIP(2) EDIT('*** MEMBER OF HOME 2 ***')
            (COL(10),A) ;
   PUT FILE (OUT12) SKIP(2) EDIT(' ')(COL(10),A) ;
   PUT FILE (OUT12) EDIT((N2(I),',' DO I = 1 TO M-2))(A) ;
   PUT FILE (OUT12) EDIT(N2(I))(A) ;
   PUT FILE (OUT12) SKIP(2) EDIT
            ('*** UNION MEMBER OF HOME1 AND HOME2 ***')
            (COL(10),A) ;
   PUT FILE (OUT12) SKIP(2) EDIT ((UN(I) DO I = 1 TO K))
            (COL(20),A) ;
   PUT FILE (OUT12) SKIP(2) EDIT
            ('*** INTERSECTION MEMBER OF HOME1 AND HOME2 ***')
            (COL(10),A) ;
   PUT FILE (OUT12) SKIP(2) EDIT ((INTER(I) DO I = 1 TO A))
            (COL(20),A) ;
   CLOSE FILE (IN121) ;
   CLOSE FILE (IN122) ;
   CLOSE FILE (OUT12) ;
END A12 ;
```

Output จะมีลักษณะดังนี้

```
*** MEMBER OF HOME 1 ***

 WONPEN,SAKDA,OPAS,KOVIT,PANEE,SOMSRI,DANG

*** MEMBER OF HOME 2 ***

 VICHAI,DANG,SOPA,SOMSRI,PANEE,WONPEN

*** UNION MEMBER OF HOME1 AND HOME2 ***

          WONPEN
          SAKDA
          OPAS
          KOVIT
          PANEE
          SOMSRI
          DANG
          VICHAI
          SOPA

*** INTERSECTION MEMBER OF HOME1 AND HOME2 ***

          DANG
          SOMSRI
          PANEE
          WONPEN
```

เฉลยข้อ 19

```
A16_4 : PROCEDURE OPTIONS(MAIN) ;
        DCL A(20) CHAR(20) ;
        DCL B(20) CHAR(20) STATIC VAR ;
        DCL (P,Q,I,N,L,C,D) FIXED BIN STATIC INIT(0) ;
        DCL (X,Y) CHAR(20) VAR ;
        DCL (SUBSTR,LENGTH) BUILTIN ;
        DCL OUT16 FILE ;
        OPEN FILE (OUT16) STREAM PRINT ;
        PUT SKIP EDIT('HOW MANY MESSAGE TO PROCESS')(A) ;
        PUT SKIP EDIT('>')(A) ;
        GET LIST(N) ;
        GET LIST((B(I) DO I = 1 TO N )) ;
        PUT FILE (OUT16) SKIP EDIT('INPUT','OUTPUT')
            (COL(15),A,X(25),A) ;
        PUT FILE (OUT16) SKIP EDIT('-----','-------')
            (COL(15),A,X(25),A) ;
        DO I = 1 TO N ;
           A(I) = B(I) ;
           DO L = 1 TO LENGTH(B(I)) ;
              IF SUBSTR(B(I),L,1) = ':' THEN C = L - 2 ;
                                        ELSE D = D + 1 ;
           END ;
           DO WHILE(D=LENGTH(B(I))) ;
              PUT FILE (OUT16) SKIP(2) EDIT(A(I),'N')
                       (COL(13),A,X(15),A) ;
              D = -9 ;
           END;
           DO WHILE(C>0) ;
              X = SUBSTR(B(I),1,C) ;
              Y = SUBSTR(B(I),C+4,LENGTH(A(I))-(C+3)) ;
              P = LENGTH(X) ;
              Q = LENGTH(Y) ;
              IF P>0
               THEN PUT FILE (OUT16) SKIP(2) EDIT(A(I),'L')
                             (COL(13),A-X(15),A) ;
               ELSE DO ;
                     IF P<Q
                       THEN PUT FILE (OUT16) SKIP(2) EDIT
                                     (A(I),'R')(COL(13),A,X(15),A) ;
                       ELSE DO ;
                             IF X ^= Y
                              THEN PUT FILE (OUT16) SKIP(2) EDIT
                                            (A(I),'D')
                                            (COL(13),A,X(15),A) ;
                              ELSE PUT FILE (OUT16) SKIP(2) EDIT
                                            (A(I),'S')
                                            (COL(13),A,X(15),A) ;
                            END ;
                     END ;
              C = 0 ;
              D = 0 ;
           END ;
        END ;
        CLOSE FILE (OUT16) ;
      END A16_4 ;
```

Output จะมีลักษณะดังนี้

INPUT	OUTPUT
SAMPLE SAMPLE	N
SHORT : LONG	L
LONG : SHORT	R
LIKE : LOVE	D
SAMPLE : SAMPLE	S

เฉลยข้อ 20

```
/* PROGRAM TO CREATE 2 ARRAYS ( EACH ARRAY HAS ONE DIMENSION ) */
          /*... NAME 'LIST_A' AND 'LIST_B' ...*/
       /*... EACH ARRAY KEEPS TEN CUSTOMERS NAME ...*/
 /*... NAME IN EACH ARRAY HAVE ARRANGE IN ASCENDING ORDER ...*/
 /*... AND THEN MERGE LIST_A AND LIST_B INTO THE NEW ARRAY ...*/
 /*... NAME 'NEW_LIST' IN ASCENDING ORDER. IF REPEAT THEN ...*/
               /*... KEEP ONLY ONE NAME ...*/

A6 : PROCEDURE OPTIONS (MAIN) ;

          /*... SUB-ROUTINE PROCEDURE PN ...*/

   PN : PROCEDURE (K,P,Q) ;
      DCL (C,K,P,Q) FIXED BIN(7) ;
      DO C = Q REPEAT(C+1) WHILE(C^>P) ;
         K = K + 1 ;
         NEW_LIST(K) = LIST_A(C) ;
      END ;
      RETURN ;
     END PN ;

          /*... SUB-ROUTINE PROCEDURE PM ...*/

   PM : PROCEDURE (K,S,T) ;
      DCL (C,K,S,T) FIXED BIN (7) ;
      DO C = T REPEAT(C+1) WHILE(C^>S) ;
         K = K + 1 ;
         NEW_LIST(K) = LIST_B(C) ;
      END ;
      RETURN ;
     END PM ;

        /*... MAIN PROGRAM ...*/

      DCL (I,J) FIXED BIN(7) STATIC INIT(1) ;
      DCL (A,K,N,M,Z) FIXED BIN(7) STATIC INIT(0) ;
      DCL (LIST_A(30),LIST_B(30)) CHAR(30) VAR STATIC ;
      DCL NEW_LIST(60) CHAR(30) VAR ;
      DCL (IN1,IN2,OUT6) FILE ;
      OPEN FILE (IN1) STREAM INPUT ;
      OPEN FILE (IN2) STREAM INPUT ;
      OPEN FILE (OUT6) STREAM PRINT ;
      PUT EDIT ('INPUT NUMBER OF ELEMENT IN LIST_A')(A) ;
      PUT SKIP EDIT ('?')(A) ;
      GET LIST (N) ;
      PUT SKIP(2) EDIT ('INPUT NUMBER OF ELEMENT IN LIST_B')(A) ;
      PUT SKIP EDIT ('?')(A) ;
      GET LIST (M) ;
```

```
      GET FILE (IN1) LIST ((LIST_A(A) DO A = 1 TO N)) ; '
      PUT FILE (OUT6) EDIT ('DATA IN LIST_A')(COL(30),A) ;
      PUT FILE (OUT6) EDIT ('**************')(COL(30),A) ;
      PUT FILE (OUT6) SKIP(2) EDIT((LIST_A(A) DO A = 1 TO N))
               (COL(30),A) ;
      GET FILE (IN2) LIST ((LIST_B(A) DO A = 1 TO M)) ;
      PUT FILE (OUT6) SKIP(2) EDIT ('DATA IN LIST_B')(COL(30),A) ;
      PUT FILE (OUT6) EDIT ('**************')(COL(30),A) ;
      PUT FILE (OUT6) SKIP(2) EDIT ((LIST_B(A) DO A =.1 TO M))
               (COL(30),A) ;
      PUT FILE (OUT6) SKIP(2) EDIT ('OUTPUT DATA IN NEW_LIST')
               (COL(27),A) ;
      PUT FILE (OUT6) EDIT ('************************')(COL(27),A)
      DO K = 1 REPEAT(K+1) WHILE(K~>M+N-Z) ;
       IF LIST_A(I) < LIST_B(J)
        THEN DO ;
              NEW_LIST(K) = LIST_A(I) ;
              I = I + 1 ;
              IF I > N THEN CALL PM(K,M,J) ;
             END ;
        ELSE DO ;
              IF LIST_A(I) > LIST_B(J)
               THEN DO ;
                     NEW_LIST(K) = LIST_B(J) ;
                     J = J + 1 ;
                     IF J > M THEN CALL PN(K,N,I) ;
                    END ;
               ELSE DO ;
                     NEW_LIST(K) = LIST_A(I) ;
                     I = I + 1 ;
                     J = J + 1 ;
                     Z = Z + 1 ;
                     IF I > N
                      THEN CALL PM(K,M,J) ;
                      ELSE IF J > M THEN CALL PN(K,N,I) ;
                    END ;
              END ;
     END ;
     PUT FILE (OUT6) SKIP(2) EDIT ((NEW_LIST(A) DO A = 1 TO K))
               (COL(30),A) ;
     CLOSE FILE (IN1) ;
     CLOSE FILE (IN2) ;
     CLOSE FILE (OUT6) ;
END A6 ;
```

OUTPUT ASSIGNMENT 6
SAMPLE RUN CASE LIST_A = LIST_B

```
DATA IN LIST_A
**************

AJCHARA   WORAVONG
ANAN      MEEMARK
BANDIT    BOONLONG
BINSAK    ALONG
BOONMEE   SRISAK

DATA IN LIST_B
**************

AKARAT    NAKPLOK
BANDIT    BOONLONG
BENJAWAN  PUTTHICHAI
DARANEE   SINGKHON
DUANGJAI  JAIRAK

OUTPUT DATA IN NEW_LIST
***********************

AJCHARA   WORAVONG
AKARAT    NAKPLOK
ANAN      MEEMARK
BANDIT    BOONLONG
BENJAWAN  PUTTHICHAI
BINSAK    ALONG
BOONMEE   SRISAK
DARANEE   SINGKHON
DUANGJAI  JAIRAK
```

SAMPLE RUN CASE LIST_A < LIST_B

```
DATA IN LIST_A
**************

AJCHARA   WORAVONG
ANAN      MEEMARK
BANDIT    BOONLONG

DATA IN LIST_B
**************

AKARAT    NAKPLOK
BANDIT    BOONLONG
BENJAWAN  PUTTHICHAI
DARANEE   SINGKHON
DUANGJAI  JAIRAK
JARAN     SANGSRI
KAMONRAT  JAIDEE

OUTPUT DATA IN NEW_LIST
***********************

AJCHARA   WORAVONG
AKARAT    NAKPLOK
ANAN      MEEMARK
BANDIT    BOONLONG
BENJAWAN  PUTTHICHAI
DARANEE   SINGKHON
DUANGJAI  JAIRAK
JARAN     SANGSRI
KAMONRAT  JAIDEE
```

SAMPLE RUN CASE LIST_A > LIST_B

```
DATA IN LIST_A
**************

AJCHARA   WORAVONG
ANAN      MEEMARK
BANDIT    BOONLONG
BINSAK    ALONG
BOONMEE   SRISAK
JAKKIT    PUTHAI
LAMAS     SUKPROM

DATA IN LIST_B
**************

AKARAT    NAKPLOK
BANDIT    BOONLONG
BENJAWAN  PUTTHICHAI

OUTPUT DATA IN NEW_LIST
***********************

AJCHARA   WORAVONG
AKARAT    NAKPLOK
ANAN      MEEMARK
BANDIT    BOONLONG
BENJAWAN  PUTTHICHAI
BINSAK    ALONG
BOONMEE   SRISAK
JAKKIT    PUTHAI
LAMAS     SUKPROM
```