

ตอนที่ 3

ตัวอย่างจากภัยพิบัติและตัวอย่าง

การวิ่งไปรวม

1. ตัวอย่างโจทย์ปัญหา

เนื่องจากนักศึกษาผ่านกระบวนการวิชา OR 213 มาแล้ว ในการเข้าห้องปฏิบัติการนั้น เมื่อนักศึกษาได้รับโจทย์ปัญหาหนึ่ง ๆ นักศึกษามีสิทธิ์จะเลือกใช้คำสั่งและวิธีการใด ๆ ที่เรียน มาแล้วในกระบวนการวิชา OR 213 ดังนั้นนอกเหนือจากคำแนะนำซึ่งอาจกำหนดให้แล้ว จะไม่ระบุ วิธีทำให้นักศึกษา

1. จงเขียนโปรแกรมเพื่อพิมพ์ตารางสูตรคูณแม่ 2 ถึงแม่ 12 ตามรูปแบบที่กำหนดให้ดังนี้ (ไม่ให้ใช้ข้อมูลเข้า จึงไม่ต้องสร้างเพิ่มข้อมูลเข้า สร้างแต่เพิ่มข้อมูลออก)

MUTIPLICATION TABLES

$$\left. \begin{array}{l} 2 * 1 = 2 \\ 2 * 2 = 4 \\ \vdots \\ 2 * 12 = 24 \end{array} \right\} \text{ไม่ต้องเว้นบรรทัด}$$

ระหว่างแม่ต่าง ๆ
ให้เว้น 1 บรรทัด

$$\begin{array}{l} \text{-->} \\ 3 * 1 = 3 \\ 3 * 2 = 6 \\ \vdots \\ 3 * 12 = 36 \\ \text{-->} \\ \\ \\ \text{-->} \\ 12 * 1 = 12 \\ 12 * 2 = 24 \\ \\ 12 * 12 = 144 \end{array}$$

2. จงเขียนโปรแกรมเพื่อพิมพ์ตารางสูตรคูณแม่ 8 ถึงแม่ 12 ตามรูปแบบที่กำหนดให้ดังนี้
(ไม่ให้ใช้ข้อมูลเข้า จึงไม่ต้องสร้างเพิ่มข้อมูลเข้า สร้างแต่เพิ่มข้อมูลออก)

MUTIPLICATION TABLES

8	**9**	**12**
8* 1 = 8	9* 1 = 9	12* 1 = 12
8* 2 = 16	9* 2 = 18	12* 2 = 24
:	:	:
8*12 = 96	9*12 = 108	12*12 = 144

3. ในการคิดค่าบริการการใช้เครื่องคอมพิวเตอร์ของศูนย์แห่งหนึ่ง สมมติว่าทางศูนย์ได้กำหนดค่าบริการการใช้เวลาเครื่องคอมพิวเตอร์ดังนี้

3000 บาทต่อเดือน (เป็นค่าวัสดุอุปกรณ์ ค่าบริการอื่น ๆ ที่ลูกค้าทุกรายต้องจ่าย) และค่าบริการรายชั่วโมงที่ใช้จริงในอัตราดังนี้

- 120 บาท/ชั่วโมง สำหรับ 10 ชั่วโมงแรก
- 90 บาท/ชั่วโมง สำหรับชั่วโมงที่ 11-20
- 70 บาท/ชั่วโมง สำหรับชั่วโมงที่ 21-50

จงเขียนโปรแกรมคำนวณค่าใช้จ่ายในการใช้เครื่องคอมพิวเตอร์จากศูนย์บริการแห่งนี้ ถ้าใช้เป็นเวลา 1-50 ชั่วโมง

กำหนดรูปแบบของข้อมูลออก

A COMPUTER SERVICE

USED HOURS	AMOUNT CHARGED (BAHT)
1	...
2	...
:	:
50	...

หมายเหตุ ไม่ต้องสร้างเพิ่มข้อมูลเข้า ให้ใช้คำสั่ง DO

4. จากแฟ้มข้อมูลของนักศึกษาที่เข้าสอบวิชา OR 205 ภาคฤดูร้อน/2535 จำนวน 30 คน (นักศึกษาร่างแฟ้มข้อมูลจากข้อมูลที่กำหนดให้) ให้นักศึกษาแต่ละคนมีเรคคอร์ดซึ่งมี format ข้างล่างนี้

ฟิลด์	คอลัมน์	รายการข้อมูล
1	1- 8	รหัสประจำตัว
2	9-40	ชื่อ-นามสกุล
3	41	เพศ: ชาย = M, หญิง = F
4	42-43	คะแนน

จงเขียนโปรแกรมเพื่อ

- 1) พิมพ์รายชื่อนักศึกษาที่เข้าสอบ รวมทั้งเกรดที่ได้ (ห้ามกำหนดเกรดใน input record)

โดยใช้เกณฑ์ดังนี้

G : คะแนน ≥ 85

P : $60 \leq$ คะแนน ≤ 84

F : คะแนน < 60

- 2) หาคะแนนเฉลี่ยของนักศึกษาที่สอบผ่านเท่านั้น
- 3) นับจำนวนนักศึกษาชายที่สอบผ่านและหาคะแนนเฉลี่ยของนักศึกษาชายกลุ่มนี้
- 4) นับจำนวนนักศึกษาหญิงที่สอบผ่านและหาคะแนนเฉลี่ยของนักศึกษาหญิงกลุ่มนี้

กำหนดรูปแบบของข้อมูลออก

OR 205

GRADE REPORT

NUMBER	STUDENT	I.D.	NAME	SEX	SCORE	GRADE
1
2
:			:	:		:
30

AVERAGE SCORE OF PASSED STUDENTS = _____ POINTS
 NUMBER OF FEMALE = _____ PERSONS
 FEMALE AVERAGE SCORE = _____ POINTS
 NUMBER OF MALE = _____ PERSONS
 MALE AVERAGE SCORE = _____ POINTS

สมมติข้อมูลเข้าดังนี้ (รหัสประจำตัว ชื่อนามสกุล ให้นักศึกษาสมมติเอาเอง)

	รหัส	เพศ	คะแนน		รหัส	เพศ	คะแนน
		M	63	16		M	61
		F	68	17		F	71
		F	56	18		M	56
		M	73	19		M	60
		M	61	20		F	62
		M	79	21		F	80
		M	53	22		M	63
	8	M	64	23		M	96
	9	M	88	24		F	86
	10	F	71	25		F	76
	11	F	64	26		M	49
1	2	F	48	27		F	54
	13	M	83	28		M	60
	14	M	71	29		M	84
	15	M	46	30		F	80

5. จากข้อมูลของอาจารย์ 25 คนในภาควิชาหนึ่ง ข้อมูลของอาจารย์ 1 คน บันทึกใน 1 เรคคอร์ด ซึ่งมี format ดังนี้

คอลัมน์	รายการข้อมูล
1 - 32	ชื่อ นามสกุล
34	เพศ (Sex) : ชาย = 1, หญิง = 2
36 - 38	อายุ (Age)
39	สถานภาพสมรส (Marital status) $\left\{ \begin{array}{l} \text{โสด} = 1 \\ \text{แต่งงานแล้ว} = 2 \\ \text{อื่น ๆ} = 3 \end{array} \right.$
41	จำนวนปีที่สอนที่มหาวิทยาลัยรามคำแหง (Years of experience)
43	ปริญญาสูงสุดที่ได้รับ (Degree) $\left\{ \begin{array}{l} \text{ตรี} = 1 \\ \text{โท} = 2 \\ \text{เอก} = 3 \end{array} \right.$

จงเขียนโปรแกรมเพื่อหา

1. อายุเฉลี่ย (average age) และพิมพ์รายชื่อผู้ที่มีอายุต่ำกว่าอายุเฉลี่ย
2. จำนวนปีที่สอนเฉลี่ย (average experience) และพิมพ์รายชื่อผู้ที่มีจำนวนปีที่สอนมากกว่าค่าเฉลี่ยนี้
3. พิมพ์ตารางแสดงจำนวนอาจารย์จำแนกตามเพศและสถานภาพสมรส รวมทั้งพิมพ์ row totals, column totals และ grand total
4. พิมพ์ตารางแสดงจำนวนอาจารย์จำแนกตามปริญญาที่ได้รับ และคำนวณเปอร์เซ็นต์ของแต่ละกลุ่มด้วย

หมายเหตุ ข้อ 1. และ ข้อ 2. คำนวณค่าให้เป็นเลขจำนวนจริง

กำหนด output format

RAMKHAMHAENG UNIVERSITY
 FACULTY OF SCIENCE
 DEPARTMENT OF STATISTICS

PERSONS WHO ARE YOUNGER THAN AVERAGE AGE, I.E., _____ YEARS.

NAME	SEX	AGE	MARITAL STATUS	YEARS OF EXPERIENCE	DEGREE
..
..

PERSONS WHO HAVE MORE EXPERIENCE THAN AVERAGE EXPERIENCE,
 I. E., _____ YEARS.

NAME	SEX	AGE	MARITAL STATUS	YEARS OF EXPERIENCE	DEGREE
..
..

TABLE 1
 NUMBER OF INSTRUCTORS CLASSIFIED
 BY
 SEX AND MARITAL STATUS

MARITAL STATUS	SEX		TOTAL
	MALE	FEHALE	
SINGLE			
MARRIED			
OTHER			
TOTAL			

TABLE 2 (ตารางนี้ให้ชนหน้าใหม่)

NUMBER OF INSTRUCTORS CLASSIFIED BY DEGREE

DEGREE	NUMBER	PERCENT
BACHELOR		
MASTER		
DOCTOR		
TOTAL		

กำหนด Input data (ชื่อ นามสกุล ให้สมมติเอาเอง)

คนที่	เพศ	อายุ	สถานภาพสมรส	จำนวนปีที่สอน	ปริญญาสูงสุด
1	ช	48	แต่งงาน	4	เอก
2	ญ	42	แต่งงาน	6	โท
3	ช	39	แต่งงาน	6	โท
4	ช	45	แต่งงาน	4	โท
5	ช	41	อื่น ๆ	4	โท
6	ช	35	อื่น ๆ	5	เอก
7	ญ	29	โสด	3	เอก
8	ช	27	โสด	3	ตรี
9	ญ	32	โสด	6	โท
10	ญ	27	โสด	2	ตรี
11	ช	25	โสด	1	ตรี
12	ช	26	โสด	1	โท
13	ญ	34	แต่งงาน	5	เอก
14	ช	29	แต่งงาน	3	โท

คนที่	เพศ	อายุ	สถานภาพสมรส	จำนวนปีที่สอน	ปริญญาสูงสุด
15	d	30	แต่งงาน	4	โท
16	d	26	โสด	2	ตรี
17	ญ	26	แต่งงาน	4	ตรี
16	ญ	32	อื่น ๆ	3	tan
19	d	30	อื่น ๆ	2	โท
20	ญ	37	อื่น ๆ	5	In
21	ญ	36	แต่งงาน	4	โท
22	d	40	แต่งงาน	8	โท
23	ช	24	โสด	1	ตรี
24	ช	45	แต่งงาน	6	tan
25	ญ	33	อื่น ๆ	3	โท

6. ฝ่ายการเงินของหน่วยงานแห่งหนึ่งต้องการคำนวณภาษีหัก ณ ที่จ่าย (ต่อเดือน)

โดยจะคำนวณจากเงินได้สุทธิต่อปี โดยที่

$$\text{เงินได้สุทธิ} = (\text{เงินเดือน} + \text{เงินพิเศษ}) \times 12$$

สมมติว่าในหน่วยงานนี้พนักงานคนที่มีรายได้สูงสุดนั้นมีรายได้ไม่เกิน 150,000 บาท

ตารางแสดงอัตราภาษี

ขั้นเงินได้สุทธิต่อปีตั้งแต่	เงินได้สุทธิจำนวนสูงสุดของขั้น	อัตราภาษีร้อยละ	ภาษีของขั้น	ภาษีสะสมสูงสุดของขั้น
1- 30,000	30,000	7	2,100	2,100
30,001- 60,000	30,000	10	3,000	5,100
60,001-100,000	40,000	13	5,200	10,300
100,001-150,000	50,000	17	8,500	18,800

ตัวอย่างการคำนวณภาษีหัก ณ ที่จ่าย (ต่อเดือน)

สมมติว่าพนักงานคนหนึ่งมีรายได้สุทธิ 32,734 บาทต่อปี

30,000 บาท เสียภาษี 7% = 2,100.00 u-in

2,734 บาท เสียภาษี 10% = 273.40 บาท

ดังนั้น 32,734 บาท เสียภาษี = 2,373.40 บาท/ปี นั่นคือเสียภาษี 197.76 บาท/เดือน

กำหนด Input record ของพนักงานคนหนึ่งดังนี้

ฟิลด์ที่	คอลัมน์	รายการข้อมูล
1	1- 5	รหัสประจำตัว (code)
2	6-25	ชื่อ นามสกุล
3	26-30	เงินเดือนต่อเดือน (salary)
4	31-34	เงินพิเศษต่อเดือน (extra money) สมมติว่าได้เท่ากันทุกเดือน

จงเขียนโปรแกรมเพื่อคำนวณภาษีหัก ณ ที่จ่ายต่อเดือนของพนักงานแต่ละคน คำนวณหาเงินเดือนรวม เงินพิเศษรวม และเงินภาษีหัก ณ ที่จ่ายรวมของพนักงานทุกคน แล้วแสดงผลตามรูปแบบที่กำหนดให้ดังนี้

DEPARTMENT AA

TAX DEDUCTION

NUMBER	CODE	NAME	SALARY/MONTH	EXTRA/MONTH	TAX/MONTH
1
2
:	:	:		:	
		TOTAL

หมายเหตุ ในการคำนวณเงินภาษีให้เก็บเศษสตางค์ไว้ด้วย

กำหนดข้อมูลเข้าของพนักงาน ดังนี้

คนที่	รหัสประจำตัว	ชื่อ นามสกุล (ให้สมมติเอง)	เงินเดือน/เดือน	เงินพิเศษ/เดือน
1	AA001		10,365	2,000
2	AA002		10,365	1,800
3	AA003		6,475	800
4	AA004		6,935	500
5	AA005		6,935	300
6	AA006		4,945	300
7	AA007		5,465	400
8	AA008		4,165	350
9	AA009		3,115	200
10	AA010		3,745	200
11	AA011		3,745	250
12	AA012		6,935	350
13	AA013		4,965	200
14	AA014		11,415	1,000
15	AA015		9,875	700

7. กำหนดข้อมูล $(x_i, y_i), i = 1, 2, \dots, 30$

i	x	y	i	x	y	i	x	y
1	10.98	35.3	11	7.82	28.9	21	8.88	72.1
2	11.13	29.7	12	0.24	28.1	22	7.68	58.1
3	12.51	30.8	13	12.19	29.1	23	8.47	44.6
4	8.40	58.8	14	11.88	46.8	24	8.86	33.4
5	9.27	61.4	15	9.57	48.5	25	10.36	28.6
6	8.73	71.3	16	10.94	59.3	26	11.08	32.7
7	6.36	74.4	17	9.58	70.0	27	13.45	57.4
8	8.50	70.7	18	10.09	70.0	28	10.00	49.7
9	8.82	57.5	19	8.11	74.5	29	12.12	34.6
10	9.14	46.4	20	6.83	71.3	30	9.20	60.9

กำหนดให้ใช้ input record 5 records ดังนี้

record ที่ 1 มีค่าของ $(x_i, y_i), i = 1, \dots, 6$

record ที่ 2 มีค่าของ $(x_i, y_i), i = 7, \dots, 12$

record ที่ 3 มีค่าของ $(x_i, y_i), i = 13, \dots, 18$

record ที่ 4 มีค่าของ $(x_i, y_i), i = 19, \dots, 24$

record ที่ 5 มีค่าของ $(x_i, y_i), i = 25, \dots, 30$

ในปัญหาของการวิเคราะห์สหสัมพันธ์ (Correlation analysis)

- สัมประสิทธิ์สหสัมพันธ์ของตัวอย่าง (Sample correlation coefficient)

$$r = S_{xy} / \sqrt{S_{xx} S_{yy}}$$

โดยที่ $S_{xy} = \sum x_i y_i - (\sum x_i)(\sum y_i)/n,$

$$S_{xx} = \sum x_i^2 - (\sum x_i)^2/n,$$

$$S_{yy} = \sum y_i^2 - (\sum y_i)^2/n, \quad \text{โดยที่ } n = 30, \Sigma = \sum_{i=1}^n$$

2. ในการทดสอบ $H_0: \rho = 0, H_1: \rho \neq 0, \alpha = .05$

เขตวิกฤติ (critical region) คือ $T > t_{\alpha/2, n-2} = 2.048$

$$ค่าสถิติที่ใช้ทดสอบคือ $t_c = r\sqrt{n-2} / \sqrt{1-r^2}$$$

ถ้า t_c ตกอยู่ในเขตวิกฤติ เราจะปฏิเสธ H_0

ถ้า t_c ตกอยู่นอกเขตวิกฤติ เราจะไม่ปฏิเสธ H_0

จงเขียนโปรแกรมเพื่ออ่านค่า x ทั้ง 30 ค่า และ y ทั้ง 30 ค่า เก็บในอะเรย์ x และอะเรย์ y ตามลำดับ แล้วคำนวณค่า r และค่า t_c ตามสูตรที่กำหนดให้ แล้วเขียนคำสั่งเพื่อตรวจสอบค่า t_c ว่าตกอยู่ในเขตวิกฤติหรือไม่ ดังนี้

ก) ถ้า t_c ตกอยู่ในเขตวิกฤติ เราจะปฏิเสธ H_0 ให้พิมพ์สรุป (conclusion) ii
REJECT NULL HYPOTHESIS (H-NULL: RHO = 0) AT ALPHA = .05

ข) ถ้า t_c ตกอยู่นอกเขตวิกฤติ เราจะไม่ปฏิเสธ H_0 ให้พิมพ์สรุปว่า
CAN NOT REJECT NULL HYPOTHESIS (H-NULL: RHO = 0) AT ALPHA = .05

กำหนด Output format

```

CORRELATION ANALYSIS
30 OBSERVATIONS
SUMMATION X = _____ <--- Σ xi
SUMMATION Y = _____ <--- Σ yi
SUM X - SQUARED (X**2) = _____ <--- Σ xi2
SUM Y - SQUARED (Y**2) = _____ <--- Σ yi2
SUM OF CROSSED PRODUCT (X*Y) = _____ <--- Σ xiyi
S(X,X) = _____ <--- Sxx
S(Y,Y) = _____ <--- Syy
S(X,Y) = _____ <--- Sxy
R (CORRELATION COEFFICIENT) = _____ <--- r
T (COMPUTED) FOR TESTING NULL HYPOTHESIS (RHO = 0) = _____ <--- tc
T(28,.025) = 2.048
CONCLUSION: _____ [ก) หรือ ข)]

```

8. นักเรียนในวิชาคณิตศาสตร์เบื้องต้นถูกแบ่งออกเป็น 3 sections แต่ละ section สอนโดยครูแต่ละคน (นักเรียนใน 3 กลุ่มมีความรู้พื้นฐานและความสามารถทางสมองพอ ๆ กันหมด) ผลการสอบได้ (โดยใช้ข้อสอบเดียวกันเป็นดังนี้)

section 1	73	89	82	43	80	73	66	60	45	93	36	77			
section 2	88	78	48	91	51	85	74	77	31	78	62	76	96	80	56
section 3	68	79	56	91	71	71	87	41	59	68	53	79	15		

ถ้าตัวอย่างสุ่มทั้ง 3 เป็นตัวอย่างจาก 3 Normal populations คือ $N(\mu_1, \sigma^2)$, $N(\mu_2, \sigma^2)$ และ $N(\mu_3, \sigma^2)$ ตามลำดับ

ในการทดสอบ

$H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu$, $H_1: \text{มี mean อย่างน้อย 1 คู่ไม่เท่ากัน}$

เราใช้ตัวสถิติทดสอบ F

ถ้าข้อมูลมีโครงสร้างดังนี้

	ค่าสังเกต	ขนาดของตัวอย่าง	Total	Mean
วิธีการที่ 1	$x_{11}, x_{12}, \dots, x_{1n_1}$	n_1	T_1	\bar{x}_1
วิธีการที่ 2	$x_{21}, x_{22}, \dots, x_{2n_2}$	n_2	T_2	\bar{x}_2
วิธีการที่ 3	$x_{31}, x_{32}, \dots, x_{3n_3}$	n_3	T_3	\bar{x}_3
		$N = \sum_{i=1}^3 n_i$	$G = \sum_{i=1}^3 T_i$	$\bar{x} = G/N$

โดยที่ x_{ij} คือค่าสังเกตตัวที่ j จากวิธีการที่ i , $i = 1, \dots, 3$ และ $j = 1, \dots, n_i$

$$T_i = \sum_{j=1}^{n_i} x_{ij}, \quad \bar{x}_i = T_i/n_i$$

กำหนดสูตร

$$C.F. = \left(\sum_{i=1}^3 \sum_{j=1}^{n_i} x_{ij} \right)^2 / N = G^2 / N$$

$$SST = \sum_{i=1}^3 \sum_{j=1}^{n_i} x_{ij}^2 - C.F., \quad df. = N - 1$$

$$SStr = \sum_{i=1}^3 T_i^2 / n_i - C.F., \quad df. = 2, \quad MStr = SStr/2$$

$$SSE = SST - SStr, \quad df. = N - 3, \quad MSE = SSE/(N-3)$$

$$f_c = MStr/MSE$$

สำหรับการทดสอบ H_0 ที่ $\alpha = .05$ เขตวิกฤติคือ $F > f_{(2, 97), .05} = 3.257$

จงเขียนโปรแกรมเพื่ออ่านข้อมูลทั้งหมด 40 ค่าคือ x_{ij} , $i = 1, \dots, 3$ และ $j = 1, \dots, n_i$ เข้าไปเก็บในอะเรย์ 2 มิติ x แล้วคำนวณหาค่า f_c จากสูตรที่กำหนดให้ แล้วเขียนคำสั่งเพื่อตรวจสอบค่า f_c ว่าตกอยู่ในเขตวิกฤติหรือไม่

ก) ถ้า f_c ตกอยู่ในเขตวิกฤติเราจะปฏิเสธ H_0 ให้พิมพ์สรุป (conclusion) ii REJECT NULL HYPOTHESIS (H-NUL: ALL MEANS ARE EQUAL) AT ALPHA = .05

ข) ถ้า f_c ตกอยู่นอกเขตวิกฤติเราจะไม่ปฏิเสธ H_0 ได้ ให้พิมพ์สรุปว่า CAN NOT REJECT NULL HYPOTHESIS (H-NUL: ALL HEANS ARE EQUAL) AT ALPHA = .05

C2345678901234567890...

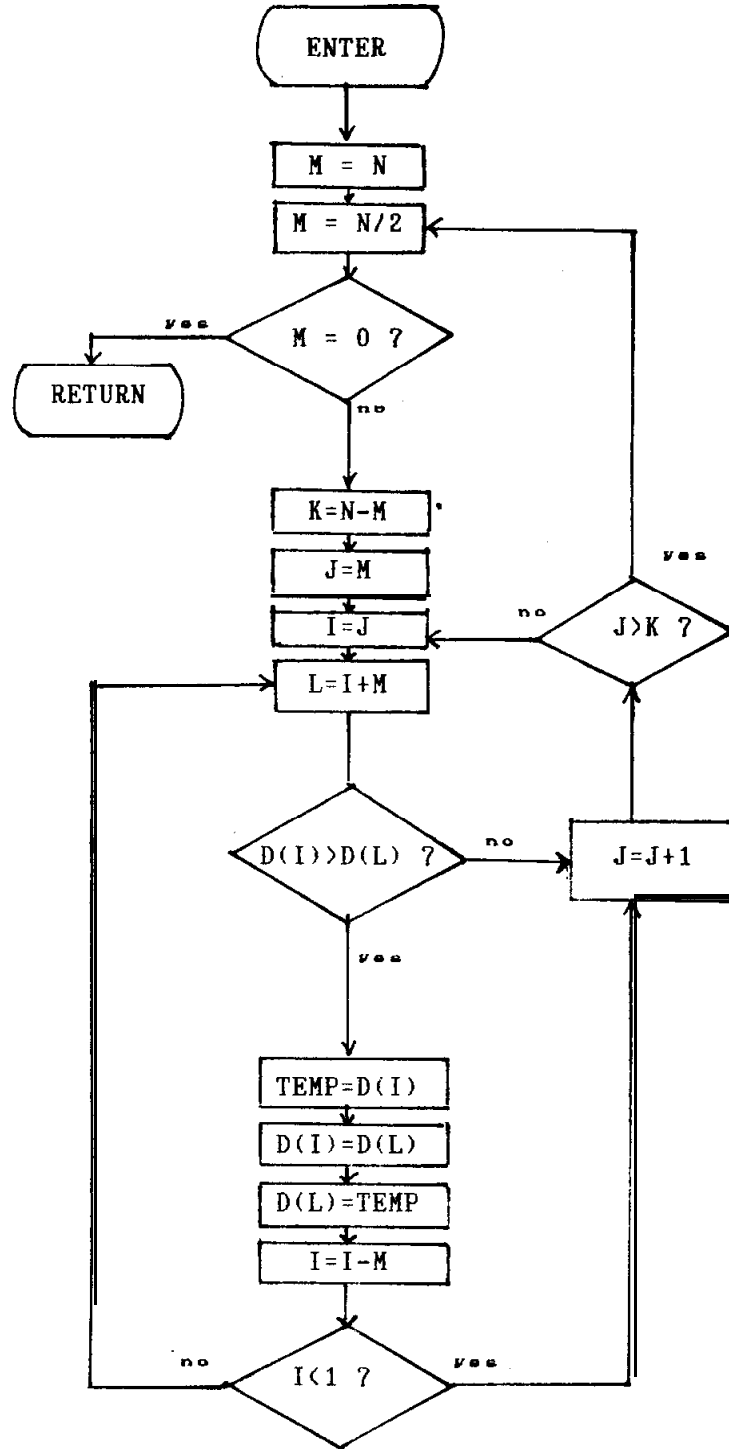
```
SUBROUTINE ASORT(A,N)
C BUBBLE SORT (DESCENDING ORDER)
C A: ONE-DIMENSIONAL ARRAY
C N: NUMBER OF ELEMENTS OF ARRAY A
DIMENSION A(N)
K = N-1
DO 16 I=1,K
L = N-I
DO 16 J=1,L
IF(A(J) .GT. A(J+1))GO TO 16
TEMP = A(J)
A(J) = A(J+1)
A(J+1) = TEHP
16 CONTINUE
RETURN
END
```

10. วิธีการเรียงลำดับอีกวิธีหนึ่งซึ่งมีประสิทธิภาพสูงคือวิธีของ Shell-Metzner หรือที่เรียกว่า shell sort ได้มีผู้ทำการเปรียบเทียบ shell sort และ bubble sort ไว้ว่า เมื่อทำการ sort เลข 100,000 จำนวน จะใช้เวลา 7.1 วัน และ 15 นาทีเมื่อใช้วิธี bubble และวิธีของ shell ตามลำดับ

จงเขียนโปรแกรมย่อยชั้บรู๊ทึน เพื่อทำการ sort ตามวิธีของ Shell-Metzner จากผังโปรแกรมที่กำหนดให้ จากนั้นให้นักศึกษาสมมติเลข 20 จำนวนขึ้นเป็นข้อมูลเข้า แล้วเขียนโปรแกรมหลักเพื่อเรียกใช้โปรแกรมย่อยชั้บรู๊ทึน แล้วพิมพ์ผลดังนี้

```
UNSORTED ARRAY: <-----10 ค่า----->
                  <-----10 ค่า----->
SORTED ARRAY:    <-----10 ค่า----->
                  <-----10 ค่า----->
```

ผังโปรแกรมเพื่อ sort สมาชิก N ตัวของอะเรย์ D



11. จากทฤษฎีทวินาม $(p+q)^n$ จะมี $(n+1)$ เทอมบวกกัน โดยที่เทอมต่าง ๆ
 คำนวณได้จาก ${}^nC_r p^r q^{n-r}$, $r=0,1,2,\dots,n$

โดยที่ ${}^nC_r = n!/[r!(n-r)!]$, $n! = n(n-1)(n-2)(n-3)\dots3.2.1$

ใน Binomial Distribution: $p+q=1$, $q=1-p$, $p=\text{Prob}(a \text{ success})$

$0 \leq p \leq 1$

ถ้า $x \sim \text{Bi}(n,p)$ แล้ว

$$\text{Pr}\{X = r\} = {}^nC_r p^r q^{n-r}, r=0,1,\dots,n$$

กำหนด โปรแกรมย่อยฟังก์ชันชื่อ FACT ซึ่งจะหาค่าของ $n!$ และ

โปรแกรมย่อยฟังก์ชันชื่อ COMB ซึ่งจะหาค่าของ nC_r ให้ดังต่อไปนี้

C234567890123

C FACTORIAL SUBPROGRAM

FUNCTION FACT(N)

FACT=1.0

DO 10 K=1,N

FN=K

10 FACT=FACT*FN

RETURN

END

C COMBINATION SUBPROGRAM

FUNCTION COMB(N,IR)

COMB=FACT(N)/(FACT(IR)*FACT(N-IR))

RETURN

END

จงเขียนโปรแกรมเพื่อสร้างตาราง binomial เมื่อ $n=10$, $p=.1(.1).9$
 โดยเรียกใช้ฟังก์ชันที่กำหนดให้ข้างต้น ในการคำนวณค่า $\text{Pr}\{X=r\}$ จากสูตรข้างต้น

ตารางข้อมูลออกมามีลักษณะดังนี้

BINOMIAL TABLES

$n = 10$

$p = \text{PROB}(A \text{ SUCCESS}) = 0.1$

$X = 0, \text{PROB}(X=0) = x.xxxx$

$X = 1, \text{PROB}(X=1) = x.xxxx$

$X=10, \text{PROB}(X=10) = x.xxxx$

$p = \text{PROB}(A \text{ SUCCESS}) = 0.2$

:

:

:

$p = \text{PROB}(A \text{ SUCCESS}) = 0.9$

$X= 0, \text{PROB}(X= 0) = x.xxxx$

$X= 1, \text{PROB}(X= 1) = x.xxxx$

$X=10, \text{PROB}(X=10) = x.xxxx$

12. ผู้จัดการฝ่ายบุคคลของห้างสรรพสินค้าแห่งหนึ่ง (มีแผนกขายสินค้าอยู่ 4 แผนก) ทางห้างซึ่งมีพนักงานขายของหน้าร้านจำนวนมาก ต้องการให้นักศึกษาช่วยเขียนโปรแกรมเพื่ออ่านและพิมพ์ข้อมูลของพนักงานทุกคน และนับจำนวนพนักงานในแผนกต่าง ๆ จำแนกตามเพศและสถานภาพสมรส

จงสร้างแฟ้มข้อมูลเข้าเพื่อเก็บข้อมูลของพนักงาน 24 คน เพื่อใช้ทดสอบการทำงานตามคำสั่งในโปรแกรมของนักศึกษา โดยที่เรคคอร์ดหนึ่งประกอบด้วย 5 รายการข้อมูล คือ รหัสประจำตัว ชื่อนามสกุล แผนก สถานภาพสมรส และเพศ (จากข้อมูลที่กำหนดให้ ให้กำหนดรูปแบบของเรคคอร์ดเอาเอง)

รหัสของข้อมูล (อ่านเป็นเลขรหัสเข้าไปเก็บ แต่เวลาพิมพ์ ให้พิมพ์เป็นข้อความ) คือ

1. แผนก(Section): 1= แผนกอุปกรณ์กีฬา (Sport)
2= แผนกเครื่องสำอาง (Cosmetic)
3= แผนกเครื่องเขียน (Stationery)
4= แผนกซูเปอร์มาร์เกต (Supermarket)

2. สถานภาพสมรส : 1= โสด (Single)
(Status) 2= แต่งงานแล้ว (Married)
3= อื่น ๆ (Others)

3. เพศ (Sex) : 1= ชาย (Male)
2= หญิง (Female)

กำหนด output format

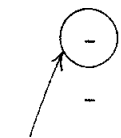
ตัวอย่าง		PERSONNEL REPORT EMPLOYEE'S RECORDS			
'CODE	NAME	SECTION	MARITAL	STATUS	SEX
101	...	SPORT		SINGLE	MALE
124	...	SUPERMARKET		OTHERS	FEMALE

ให้พิมพ์เป็นข้อความ

TABLE 1

NUMBER OF EMPLOYEES CLASSIFIED BY SECTION, MARITAL STATUS,
AND SEX

SECTION	SINGLE		MARRIED		OTHERS		TOTAL
	MALE	FEMALE	MALE	FEMALE	MALE	FEMALE	
SPORT							
COSMETIC	-						
STATIONERY	-				-	-	
SUPERMARKET	-	-			-	-	
TOTAL	-	-			-	-	



subscript คือ
(3,3,2)

- ข้อแนะนำ - ให้ใช้ระยะ 3 มิติ ซึ่งสมาชิกจะแทนเซลล์ใน TABLE 1 โดยที่
- subscript ตัวแรก แทน แผนก
 - subscript ตัวที่ 2 แทน สถานภาพสมรส
 - subscript ตัวที่ 3 แทน เพศ
- การพิมพ์ TABLE 1 ให้ใช้ implied DO
 - ในการพิมพ์ EMPLOYEE'S RECORDS ตามที่กำหนดนั้น ใช้คำสั่ง CHARACTER กับ ชื่อตามสกุล แผนก สถานภาพสมรส และ เพศ

ตัวอย่าง DIMENSION SECTION(4), STATUS(3), SEX(2)
CHARACTER SECTION*11, STATUS*7, SEX*6, NAME*30
DATA SEX/'MALE', 'FEMALE'/
DATA STATUS/'SINGLE', 'MARRIED', 'OTHERS'/

format code ในการพิมพ์ SECTION(I) ใช้ A11 ได้ทันที
STATUS(I) ใช้ A7 ได้ทันที
และ SEX(I) ใช้ A6 ได้ทันที

กำหนด input data

รหัสประจำตัวพนักงานใช้ 101,102,103,...,124 ชื่อนามสกุลตัวเอง
(เวลาตั้งคูปเพศที่กำหนดให้ประกอบด้วย)

ข้อมูลต่อไปนี้คือ แผนก สถานภาพสมรส และ เพศ ตามลำดับ

- 1) 1 1 1 6) 1 2 1 11) 2 2 2 16) 3 2 2 21) 4 3 1
 2) 1 1 1 7) 2 1 2 12) 2 2 2 17) 3 3 1 22) 4 3 1
 3) 1 1 2 8) 2 1 2 13) 3 2 1 18) a 3 1 23) 4 3 2
 4) 1 1 2 9) 2 2 1 14) 3 2 1 19) 4 2 2 24) 4 3 2
 5) 1 2 1 10) 2 2 1 15) 3 2 2 20) 4 2 2

13. กำหนดข้อมูล $(x_i, y_i), i = 1, \dots, 30$

	x	y		x	y		x	y
1.	10.98	35.3	11.	7.82	28.9	21.	8.88	72.1
2.	11.13	29.7	12.	8.24	28.1	22.	7.68	58.1
3.	12.51	30.8	13.	12.19	29.1	23.	8.47	44.6
4.	8.40	58.8	14.	11.88	46.8	24.	8.86	33.4
5.	9.27	61.4	15.	9.57	48.5	25.	10.36	28.6
6.	8.73	71.3	16.	10.94	59.3	26.	11.08	32.7
7.	6.36	74.4	17.	9.56	70.0	27.	13.45	57.4
8.	8.50	70.7	18.	10.09	70.0	28.	10.00	49.7
9.	8.82	57.5	19.	8.11	74.5	29.	12.12	34.6
10.	9.14	46.4	20.	6.83	71.3	30.	9.20	60.9

กำหนดให้ใช้ input record 5 เรคคอร์ดดังนี้

เรคคอร์ดที่ 1 มีค่าของ $(x_i, y_i), i = 1, \dots, 6$

เรคคอร์ดที่ 2 มีค่าของ $(x_i, y_i), i = 7, \dots, 12$

เรคคอร์ดที่ 3 มีค่าของ $(x_i, y_i), i = 13, \dots, 16$

เรคคอร์ดที่ 4 มีค่าของ $(x_i, y_i), i = 19, \dots, 24$

เรคคอร์ดที่ 5 มีค่าของ $(x_i, y_i), i = 25, \dots, 30$

ในปัญหาของการวิเคราะห์สหสัมพันธ์ (Correlation Analysis)

1. ส.ป.ส. สหสัมพันธ์ของตัวอย่าง (Sample correlation coefficient) คือ

$$r = S_{xy} / \sqrt{S_{xx} S_{yy}}$$

โดยที่ $S_{xx} = \sum x_i^2 - (\sum x_i)^2 / n$

$$S_{yy} = \sum y_i^2 - (\sum y_i)^2 / n$$

$$S_{xy} = \sum x_i y_i - (\sum x_i \sum y_i) / n$$

30

โดยที่ $n = 30, \sum_{i=1}^n = \sum$

2. ในการทดสอบ $H_0: \rho = 0, H_1: \rho \neq 0, \alpha = .05$

เขตวิกฤต (Critical region) คือ $|T| > t_{28, .025} = 2.048$

ตัวสถิติทดสอบคือ $t_c = r(n-2)^{1/2} / (1-r^2)^{1/2}$

ถ้า t_c ตกอยู่ในเขตวิกฤต เราจะ Reject H_0

ถ้า t_c ไม่ตกอยู่ในเขตวิกฤต เราจะไม่ Reject H_0

13.1 จงเขียนโปรแกรมย่อยซับรูทีน (Subroutine subprogram) เพื่อหาค่าส.ป.ส.

สหสัมพันธ์ (r) ของข้อมูล n observations ใดๆ นั่นคือ $(x_i, y_i), i=1, \dots, n$

ในซับรูทีนจากอะเรย์ 1 มิติ 2 อะเรย์ คือ อะเรย์ X และ Y (ซึ่งต่างก็มีสมาชิก n ตัว) ให้คำนวณและพิมพ์ค่า ดังนี้ $\sum x_i, \sum y_i, \sum x_i^2, \sum y_i^2, \sum x_i y_i, S_{xx}, S_{yy}, S_{xy}, r_{xy}$ และ t_c ทั้งนี้ค่าอภิบาลจะคงเดิมไม่ว่าจะส่งข้อมูลชุดใดจาก main program มาคำนวณในโปรแกรมย่อยซับรูทีนนี้

นักศึกษาอาจกำหนด

ขนาดของอะเรย์ X และ Y

SUBROUTINE CORR(X,Y,N,SUMX,SUMY,SUMXX,SUMYY,SUMXY,SXX,SY,
*SKY,R,TC)

13.2 จงเขียนโปรแกรมหลัก (Main program) เพื่ออ่านข้อมูลที่ละชุด รวม 3 ชุด แล้วทำการวิเคราะห์สหสัมพันธ์ของแต่ละชุด ให้เรียกใช้ซับรูทีนในข้อ 13.1 เพื่อหาและพิมพ์ค่าต่างๆ ของแต่ละชุด

ข้อมูล 3 ชุดคือ (ใช้แฟ้มข้อมูลเข้าซึ่งประกอบด้วย 5 เรคคอร์ดจากข้อ 13.1)

ชุดที่ 1 30 observations (จาก 5 เรคคอร์ด) เรียกชื่อในโปรแกรมหลักว่า

$(a_i, b_i), i=1, \dots, 30$

ชุดที่ 2 12 observations (จาก 2 เรคคอร์ดแรก) เรียกว่า

$(c_i, d_i), i=1, \dots, 12$

ชุดที่ 3 18 observations (จาก 3 เรคคอร์ดหลัง) เรียกว่า

$(e_i, f_i), i=1, \dots, 18$

ถ้าใช้คำสั่ง OPEN กำหนดแฟ้มข้อมูลเข้าเป็น unit no. 5 ก่อนอ่านข้อมูลชุดที่ 2 ให้ใช้คำสั่ง REWIND(5) เพื่อกลับไปเรคคอร์ดแรกใหม่ แล้วอ่าน 2 เรคคอร์ดแรกเป็นข้อมูลชุดที่ 2 สำหรับข้อมูลชุดที่ 3 อ่าน 3 เรคคอร์ดที่เหลือ

หมายเหตุ ในการพิมพ์จำนวน observations จะต้องเปลี่ยนจำนวนด้วยเมื่อพิมพ์

output ของแต่ละชุด ค่าตาราง t และผลสรุปจะเปลี่ยนไปด้วย

ชุดที่ 1 $N=30, t_{29, .025} = 2.048$

ชุดที่ 2 $N=12, t_{10, .025} = 2.228$

ชุดที่ 3 $N=18, t_{16, .025} = 2.120$

กำหนด output format

สำหรับแต่ละชุดของข้อมูลให้พิมพ์ตามรูปแบบนี้

CORRELATION ANALYSIS

SET (a) : (b) OBSERVATIONS

X(I) :

Y(I) :

X(I) :

Y(I) :

X(I) :

Y(I) :

พิมพ์บรรทัดละ 10 ค่า
ดังนั้น ข้อมูลชุดที่ 1 จะมี 6 บรรทัด
ชุดที่ 2 จะมี 4 บรรทัด
ชุดที่ a จะมี 4 บรรทัด

I

SUMMATION X =

SUMMATION Y =

SUM X-SQUARED (X**2) =

SUM Y-SQUARED (Y**2) =

SUM OF CROSSED PRODUCTS (X*Y)

S(X, X) =

S(Y, Y) =

S(X, Y) =

R(CORRELATION COEFFICIENT)

T(COMPUTED) FOR TESTING HYPOTHESIS (RHO=0) =

T((c), .025)

CONCLUSION: [(a) หรือ (b)]

หมายเหตุ (a) คือ ข้อมูลชุดที่ (1 or 2 or 3)

(b) คือ จำนวนข้อมูลหรือค่าของ n (30 or 12 or 18)

(c) คือ degree of freedom เท่ากับ n-2 (28 or 10 or 16)

14. โรงงานอุตสาหกรรม ABC ผลิตสินค้า มีแฟ้มข้อมูลซึ่งประกอบด้วย 9
 เรคคอร์ด ซึ่งแต่ละเรคคอร์ดมีรายการข้อมูลดังต่อไปนี้

1. หมายเลขประจำแผนก (Department number)
2. หมายเลขประจำสินค้า (Item number) ซึ่งปรากฏครั้งเดียวเท่านั้น
 ในแต่ละแผนก
3. ปริมาณสินค้า (Quantity) หน่วยเป็นชิ้น
4. ต้นทุนต่อชิ้น (Cost per item)

จงเขียนโปรแกรมเพื่อทำรายงานสรุปในลักษณะดังต่อไปนี้

ABC PLANT				
INVENTORY REPORT				
DEPARTMENT NO.	ITEM NO.	QUANTITY	COST PER ITEM	VALUE TOTALS
15	176	10	3.00	30.00
15	376	4	3.20	12.80
15	476	2	7.00	14.00
				56.80
16	276	8	2.00	16.00
16	376	5	2.50	12.50
				28.50
19	276	10	1.25	12.50
19	376	100	0.03	3.00
19	476	20	4.00	80.00
19	576	4	16.00	64.00
				159.50
			GRAND TOTAL	244.80

เป็นข้อมูลในแฟ้มข้อมูลให้นักศึกษากำหนดรูปแบบของเรคคอร์ดเอง
 (ในทึน VALUE = QUANTITY คูณ COST PER ITEM)

15. จงสร้างแฟ้มข้อมูลเข้าเพื่อเก็บยอดขายของเครื่องดื่มแต่ละชนิดในแต่ละวันจากตารางที่กำหนดให้ และจงเขียนโปรแกรมเพื่อหายอดขายของเครื่องดื่มแต่ละชนิดหายอดขายของแต่ละวัน หายอดขายรวมของทั้งอาทิตย์ หว่าเครื่องดื่มชนิดใดและขายในวันใดมียอดขายสูงสุดและหาว่าเครื่องดื่มชนิดใดและขายในวันใดมียอดขายต่ำสุด

กำหนดให้พิมพ์ผลตามรูปแบบดังนี้

XYZ DEPARTMENT STORE

DAY	SOFT DRINK				TOTAL
	KOKA-KOLA	SIX-UP	WHITE-SPOT	PEPSI-KOLA	
SUNDAY	67	39	20	58	
MONDAY	50	35	22	62	
TUESDAY	58	40	20	60	
WEDNESDAY	77	36	24	64	
THURSDAY	66	32	21	57	
FRIDAY-	70	30	25	63	
SATURDAY	40	10	15	30	
TOTAL					!

THE MAXIMUM SOLD IS (ชนิด) ON (วัน)

THE MINIMUM SOLD IS (ชนิด) -- ON (วัน)

หมายเหตุ ให้ใช้อะเรย์ 2 มิติ ชนิด CHARACTER เก็บชื่อวันและชื่อเครื่องเดิม โดยที่สมาชิกของแต่ละอะเรย์เราอาจกำหนดให้สามารถเก็บจำนวนตัวอักษรได้เท่า ๆ กัน เพื่อความสะดวกในการเขียนคำสั่ง เช่นอาจเก็บชื่อวันในอะเรย์ DAY และให้สมาชิกแต่ละตัวมีความยาว 9 bytes (ชื่อวันยาวที่สุดคือ WEDNESDAY) ในการกำหนดชื่อวันให้เก็บใน DAY แต่ละตัวใช้คำสั่ง DATA ได้ และในการพิมพ์ DAY(I) ใช้ Aw format code

การขีดเส้นตาราง ไม่ต้องขีดเส้นตั้ง สำหรับเส้นนอน ถ้าต้องการขีดด้วยสัญลักษณ์ * 100 ตำแหน่ง นักศึกษาอาจใช้ 100(1H*) หรือ 100('*')

16. ในการ update Master file หนึ่งซึ่งประกอบด้วย 10 เรคคอร์ด สมมติว่าเมื่อสิ้นสุดวันหนึ่ง รวบรวมรายการเปลี่ยนแปลงได้เป็น Transaction file ซึ่งประกอบด้วย 7 เรคคอร์ด

กำหนดรูปแบบของเรคคอร์ด (สำหรับทั้ง 2 files)

ฟิลด์	คอลัมน์	รายการข้อมูล
1	1-3	รหัสประจำของสินค้า (Item number)
2	5-6	จำนวน (Quantity) (ชิ้น) -ใน Master file คือจำนวนของ ใน stock -ใน Transaction file คือ จำนวนที่ขายได้

ตัวอย่าง Input data	ตัวอย่าง Output ที่ต้องการ
<p>ใน Master file</p> <pre> เรคคอร์ด 1 444 40 2 111 30 3 222 15 4 134 20 5 353 5 </pre>	<p>OLD INVENTORY</p> <pre> ITEM NUMBER QUANTITY 444 40 111 30 222 15 134 20 353 5 </pre>
<p>ใน Transaction file</p> <pre> เรคคอร์ด 1 222 10 2 134 3 3 111 29 4 112 9 5 353 2 6 352 12 </pre>	<p>TRANSACTIONS</p> <pre> ITEM NUMBER QUANTITY 222 10 134 3 111 29 112 9 353 2 352 12 </pre> <p>NEW INVENTORY</p> <pre> INVALID ITEM NUMBER IS 112 INVALID ITEM NUMBER IS 352 ITEM NUMBER QUANTITY 444 40 111 1 *REORDER* 222 5 *REORDER* 134 17 353 3 *REORDER* </pre>

2.1 จงเขียนโปรแกรมย่อยที่รับรู้นี้เพื่อ sort เรคคอร์ดในแฟ้มหนึ่ง โดยที่เรคคอร์ดหนึ่งมี 2 ฟีลด์ และฟีลด์ แรกเป็น key field ให้ sort เรคคอร์ด โดยให้ key field เรียงจากน้อยไปหามาก (ascending order)

2.2 จงเขียนโปรแกรมเพื่อเรียกใช้ SUBROUTINE ใน 2.1 เพื่อ sort
 เรคคอร์ดในแฟ้มทั้งสองในใจย์ ให้เรียงตามลำดับรหัสประจำของสินค้า
 จากน้อยไปหามาก แล้ว update Master file พิมพ์ผลทำนองเดียวกับข้อ 1

ตัวอย่าง Input data	ตัวอย่าง Output ที่ต้องการ
<p><u>ใน Master file</u></p> <pre> เรคคอร์ด 1 444 40 2 111 30 3 222 15 4 134 20 5 353 5 </pre>	<pre> OLD INVENTORY ITEM NUMBER QUANTITY 111 30 134 20 222 } sorted 15 353 5 444 40 </pre>
<p><u>ใน Transaction file</u></p> <pre> เรคคอร์ด 1 222 10 2 134 3 3 111 29 4 112 9 5 353 2 6 352 12 </pre>	<pre> TRANSACTIONS ITEM NUMBER QUANTITY 111 29 112 9 134 } sorted 3 352 10 353 12 352 2 </pre>
	<pre> NEW INVENTORY INVALID ITEM NUMBER IS 112 INVALID ITEM NUMBER IS 352 ITEM NUMBER QUANTITY 111 1 *REORDER* 134 17 222 5 *REORDER* 353 3 *REORDER* 444 40 </pre>

17. การวาดกราฟ $y = f(x)$

กำหนดโปรแกรมย่อยที่ชื่อว่า PLOT ซึ่งทำการพลอต 1 จุดเมื่อถูกเรียกใช้ 1 ครั้ง ในแต่ละบรรทัดส่วนที่ใช้พลอตจะใช้ 100 ตำแหน่งเพื่อพิมพ์ผล (100A1) และมีจุดที่จะถูกพลอต 1 จุดใน 1 บรรทัด

C23456789

C SUBPROGRAM TO PLOT A POINT AT A TIME

SUBROUTINE PLOT(X,Y,YMAX,YMIN)

C YMAX = MAXIMUM VALUE OF Y

C YMIN = MINIMUM VALUE OF Y

DIMENSION IPOINT(100)

CHARACTER ISTAR,IBLANK,IPOINT

DATA ISTAR,IBLANK/'*', ' ' /

DO 4 I = 1,100 \ 1 blank

4 IPOINT(I) = IBLANK

YP = (Y-YMIN)/(YMAX-YMIN)

IX = 99.0*YP+1.0

IPOINT(IX) = ISTAR

WRITE(6,11)X,Y,IX,(IPOINT(I),I=1,100)

11 FORMAT(2X,F4.1,E15.7,I5,2X,'I',100A1)

RETURN

END

↑
ใช้ขีดแกน X (แกนตั้ง)

หมายเหตุ YP มีค่าอยู่ระหว่าง 0 ถึง 1

IX มีค่า 1,2,3,...,100 (ตำแหน่งที่จะพลอตจุด)

ถ้า $Y = YMIN$, $YP = 0$, $IX = 1$

ถ้า $Y = YMAX$, $YP = 1$, $IX = 100$

ให้นักศึกษาเก็บโปรแกรมย่อยนี้ในแฟ้ม SUB.FOR

1. ตัวอย่างโปรแกรมหลัก เรียกใช้ PLOT เพื่อพลอตกราฟ

```
Y = X ** 2 ,    -2.0 <= x <= 3.0
X = 3.0 , YMAX = 9.0
x = 0.0 . YMIN = 0.0
```

C23456789

```
C TO PLOT CURVE Y = X ** 2 , X = (-2(0.1)3)
OPEN(6,FILE='MAIN.OUT',STATUS='NEW')
x = -2.0
DO 9 I = 1,51
Y = x*x
CALL PLOT(X,Y,9.0,0.0)
9 X=X+0.1
STOP
END
```

X = -2., -1.9, ..., 2.9, 3
51 จุด

ให้นักศึกษาเก็บโปรแกรมหลักนี้ใน MAIN.FOR

ขั้นตอนการคอมไพล์และลิงก์โปรแกรม

การคอมไพล์และลิงก์โปรแกรมใน MAIN.FOR และ SUB.FOR ให้ใช้คำสั่ง

\$FOR/LIS MAIN+SUB จะได้แฟ้ม MAIN.LIS และ MAIN.OBJ

(ถ้าไม่มี Compile-time errors)

ใน MAIN.LIS จะมีทั้งโปรแกรมหลักและ

โปรแกรมย่อย

\$LINK MAIN

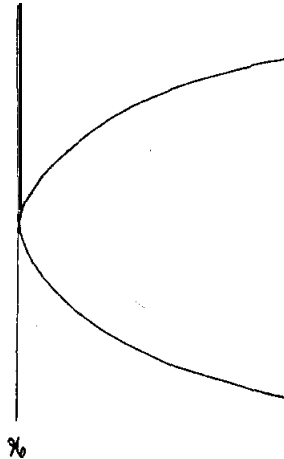
\$RUN MAIN

ให้พิมพ์ MAIN.LIS และ MAIN.OUT สิ่งด้วย

กราฟที่ได้จะอยู่ในลักษณะดังนี้ (ใน output จะมีแค่จุดที่พลอตด้วย *)

X จะอยู่เป็นแกนตั้ง

Y จะอยู่เป็นแกนนอน



2. จงเขียนโปรแกรมหลักเรียกใช้ PLOT ^{กราฟของกรวยแจกแจงแวน} เพื่อพลอต/standard normal
กราฟซึ่งมีสมการ

$$Y = (1/2\pi)^{1/2} \exp(-z^2/2) \quad (\pi = 3.141593,$$

ในที่นี้ให้พลอตจุดสำหรับค่า $z = (-2.7(0.1)2.7)$ นั่นคือพลอตทั้งหมด

5 5 จุด ($z = -2.7, -2.6, \dots, 0, \dots, 2.6, 2.7$)

└──────────────────────────────────┘

5 5 ค่า

ให้นักศึกษาเขียนคำสั่งเพื่อคำนวณหา YMAX (เมื่อ $z = 0$) และ
YMIN (เมื่อ $z = -2.7$ หรือ 2.7) จากสมการที่กำหนดให้เอาเอง ก่อนเรียกใช้
ขั้วรับที่ PLOT ให้พิมพ์ output ต่อไปนี่ก่อน [โดยที่ $e^x = \text{EXP}(X)$]

18. จงเขียนโปรแกรมคำนวณหาพื้นที่ใต้กราฟ $y = 1/(1 + x^2)$ เมื่อ $0 \leq x \leq 1$

จาก Simpson's 1/3 Rule

$$\int_a^b Y dx = (h/3)(y_0 + 4y_1 + 2y_2 + 4y_3 + \dots + 4y_{n-1} + y_n)$$

โดยที่ $h = (b - a)/n$ และ n เป็นเลขคู่

ดังนั้นถ้า $n = 100$, $h = (1 - 0)/100 = .01$

นั่นคือ x มีค่าจาก 0 ถึง 1 โดยมีค่าเพิ่มขึ้นทีละครั้งละ .01

$$y_0 = 1/(1 + 0^2) = 1, y_1 = 1/(1 + x_1^2), \dots, y_{100}$$

ไม่ต้องสร้าง input file และให้พิมพ์ผลดังนี้

```
AREA UNDER CURVE Y = 1/(1 + X**2), X FROM 0.0 TO 1.0
BY SIMPSON'S 1/3 RULE, AREA = _____
```

19. โปรแกรมย่อยชั้บรู่ที่ RANDOM เป็นตัวสร้างเลขสุ่ม (Random number) RANO ซึ่ง $0 < RANO < 1$, RANO เป็นตัวเลขสุ่มซึ่งจะมีการแจกแจงแบบยูนิฟอร์ม (Uniform distribution)

ในการเรียกใช้ชั้บรู่ที่ RANDOM แต่ละครั้ง เราต้องกำหนดค่าของ IX ซึ่งเรียกว่า "seed" โดยให้ IX มีค่าเป็นเลขคี่ใด ๆ ที่มีค่าน้อยกว่า $2^{31}-1 = 2147483647$ (ซึ่งคือค่าสูงสุดของเลขจำนวนเต็มในทษขนาด 32bit/word)

19.1 ในตัวอย่างต่อไปนั้ โปรแกรมหลักเรียกใช้ชั้บรู่ที่ RANDOM 1000 ครั้ง แต่ละครั้งได้ค่าของ RANO 1 ค่า เราอาจทำการจำลองการโยนเหรียญ 1 อัน 1000 ครั้ง แล้วนับจำนวนหัวและก้อยที่จะเกิดจากการโยนแต่ละครั้ง

โดยที่ถ้า RANO < .5 ให้ถือว่าเป็นหัว (head)

ถ้า RANO >=.5 ให้ถือว่าเป็นก้อย (tail)

ให้นักศึกษาลองวิ่งโปรแกรมนี้ (วิ่งบน MICRO VAX II, ไม่ต้องส่งข้อนี้)

โดยในตอนคอมไพล์โปรแกรม ให้ใช้คำสั่งดังนี้

```
$FOR yourfile/LIS/NOCHECK
```


↓
เพราะซึ่บรู๊ทึน RANDOM ต้องใช้การเกิด
overflow ในเลขจำนวนเต็ม ถ้าเราไม่ใส่
NOCHECK ในตอนวิ่งโปรแกรม จะเกิด
run-time-error คือจะได้ข้อความว่าเกิด
overflow ขึ้นใน integer constant

```
C2345678901234567890
```

```
C      COIN TOSSING
```

```
      INTEGER HEADS, TAILS
```

```
      OPEN(6, FILE='COIN.OUT', STATUS='NEW')
```

```
      HEADS=0 ให้นักศึกษาดังข้อใหม่เอาเองได้
```

```
      TAILS=0
```

```
      IX=4999997 <----- "seed" (ให้ใช้ seed นี้ในข้อ 19.2 ด้วย)
```

```
      DO 100 I =1, 1000
```

```
      CALL RANDOM(IX, IY, RANO)
```

```
      IF(RANO .LT. 0.5) HEADS=HEADS+1
```

```
      IF(RANO .GE. 0.5) TAILS=TAILS+1
```

```
100    IX=IY <---- เพื่อให้ IX เปลี่ยนค่าไปในการเรียก RANDOM ครั้งต่อไป
```

```
      WRITE(6, 200) HEADS, TAILS
```

```
200    FORMAT(5X, 'RESULTS OF' COIN TOSS EXPERIMENT'//
```

```
      18X, '# OF HEADS = ', I4/
```

```
      28X, '# OF TAILS = ', I4)
```

```
      STOP
```

```
      END
```

C2345678901234567890

```
C      SUBROUTINE FOR GENERATING A UNIFORM RANDOM NUMBER
      SUBROUTINE RANDOM(IX, IY, RANO)
      IY=IX*65539
      IF(IY)5,6,6
5     IY=IY+2147483647+1
6     RANO=IY
      RANO=RANO*0.4656613E-9
      RETURN
      END
```

19.2 จากซึบรoutines RANDOM ซึ่งให้เลขสุ่มที่มีการแจกแจงแบบยูนิฟอร์ม มีค่าจาก 0 ถึง 1 ถ้าเราต้องการตัวเลขสุ่มจาก 0 ถึง 9 (0,1,2,...,9) เราอาจทำได้โดยการใช้คำสั่ง

$IN = RANO * 10.0$

↑
ตัวแปรชนิด integer จะเห็นว่า IN มีค่า 0,1,...,9
และ IN จะมีการแจกแจงแบบยูนิฟอร์มด้วย

ให้นักศึกษาเขียนโปรแกรมสร้างเลขสุ่ม 1 หลัก 2000 ตัว (โดยเรียกใช้ซึบรoutines RANDOM ช่วย ให้ใช้ค่า IX = 4999997) นับจำนวนครั้งของการเกิดเลข 0, เลข 1, ..., เลข 9 เก็บจำนวนนับซึ่งเป็นความถี่ที่สังเกตได้ (observed frequency) ในอะเรย์ F ซึ่งมีสมาชิก 10 ตัวคือ F(I), I=1, ..., 10 ถ้า IN มีการแจกแจงแบบยูนิฟอร์มจริง โอกาสที่ IN จะมีค่าเป็น 0,1, ..., 9 จะมีค่าเท่า ๆ กันคือ 1/10. ดังนั้นความถี่ที่คาดหวัง (expected frequency) ของจำนวนครั้งของการเกิดค่าต่าง ๆ ของ IN คือ $2000 * (1/10.) = 200$ ครั้ง สมมติให้เป็นค่าของ E(I), I=1, ..., 10

การทดสอบว่า IN มีการแจกแจงเป็นแบบยูนิฟอร์มหรือไม่นั้น เราใช้ Chi-square test

นักศึกษาคำนวณหาค่าของ $\chi^2_c = \sum_{i=1}^{10} [(F_i - E_i)^2 / E_i] = \sum_{i=1}^{10} [(F_i - 200)^2 / 200]$

$(E_i = 200 \text{ สำหรับทุกค่าของ } i = 1, \dots, 10)$

เปรียบเทียบค่าของ χ^2_c กับ $\chi^2_{9, .05} = 16.919$ แล้วสรุปผลการทดสอบดังนี้

ก) ถ้า $\chi^2_c \geq 16.919$ เราสรุปว่าตัวเลขสุ่มที่สร้างจากซบรูทีน RANDOM ไม่มีการแจกแจงเป็นแบบยูนิฟอร์ม ให้พิมพ์สรุปว่า

CONCLUSION: RANDOM NUMBERS FROM SUBROUTINE 'RANDOM'
ARE NOT UNIFORM.

ข) ถ้า $\chi^2_c < 16.919$ เราสรุปว่าตัวเลขสุ่มดังกล่าวมีการแจกแจงแบบยูนิฟอร์ม ให้พิมพ์สรุปว่า

CONCLUSION: RANDOM NUMBERS FROM SUBROUTINE 'RANDOM'
ARE UNIFORM.

กำหนดให้พิมพ์ผลดังนี้

GOODNESS OF FIT TEST

DIGIT	0	1	2	3 . . .	9	TOTAL
F(I)	-	-	-	. . .	-	2000
E(I)	200	200	200	. . .	200	2000

CHI-SQUARE (COMPUTED) = -----

CHI-SQUARE(9, .05) = 16.919

CONCLUSION: ก) หรือ ข)

20. จากผังโปรแกรมที่กำหนดให้ จงเขียนโปรแกรมย่อยซึ่บรู๊ตึ้นชื่อ MERGE เพื่อรวมอะเรย์ A (ซึ่งมีสมาชิก N ตัว) และอะเรย์ B (ซึ่งมีสมาชิก M ตัว) เข้าด้วยกัน ให้อะเรย์ใหม่คือ X (ซึ่งมีสมาชิก L ตัว $L = N + M$) และจาก subroutine ในข้อ 10. (สมมติชื่อ SHSORT) ซึ่งทำการเรียงลำดับสมาชิกของอะเรย์ 1 มิติ จากนั้นอธิบายไปหามา

จงเขียนโปรแกรมหลักเพื่ออ่านค่าจาก file.DAT เข้าไปเก็บในอะเรย์ X และอะเรย์ Y เรียก SHSORT เพื่อเรียงลำดับสมาชิกของอะเรย์ X และเรียงลำดับสมาชิกของอะเรย์ Y แล้วเรียกใช้ซึ่บรู๊ตึ้น MERGE เพื่อรวมอะเรย์ทั้งสองเข้าด้วยกัน เรียกอะเรย์ใหม่ซึ่งเกิดจากการรวมว่า Z

กำหนด file.DAT

ให้มี 5 เรคคอร์ด เรคคอร์ดละ 10 จำนวน

อ่านค่าของอะเรย์ X จาก 2 เรคคอร์ดแรก

อ่านค่าของอะเรย์ Y จาก 3 เรคคอร์ดหลัง

ให้กำหนดเลขจำนวนจริงทั้ง 50 จำนวนเอาเอง (โดยที่แต่ละจำนวนต้องเป็นเลขที่มีทศนิยม 2 หลัก เลขตัวเต็มให้กำหนด 2 หลักขึ้นไป)

กำหนด output format,

SORT - MERGE

UNSORTED ARRAY X : <-----10 d-i----->
<-----10 ค่า----->

UNSORTED ARRAY Y : <-----10 ค่า----->
<-----10 ค่า----->
<-----10 ค่า----->

SORTED ARRAYS		NEW - ARRAY
X	Y	Z
X ₁	y ₁	Z ₁
X ₂₀	y ₂₀	Z ₂₀
	y ₂₁	Z ₂₁
	y ₃₀	Z ₃₀
		Z ₃₁
		Z ₅₀

หมายเหตุ

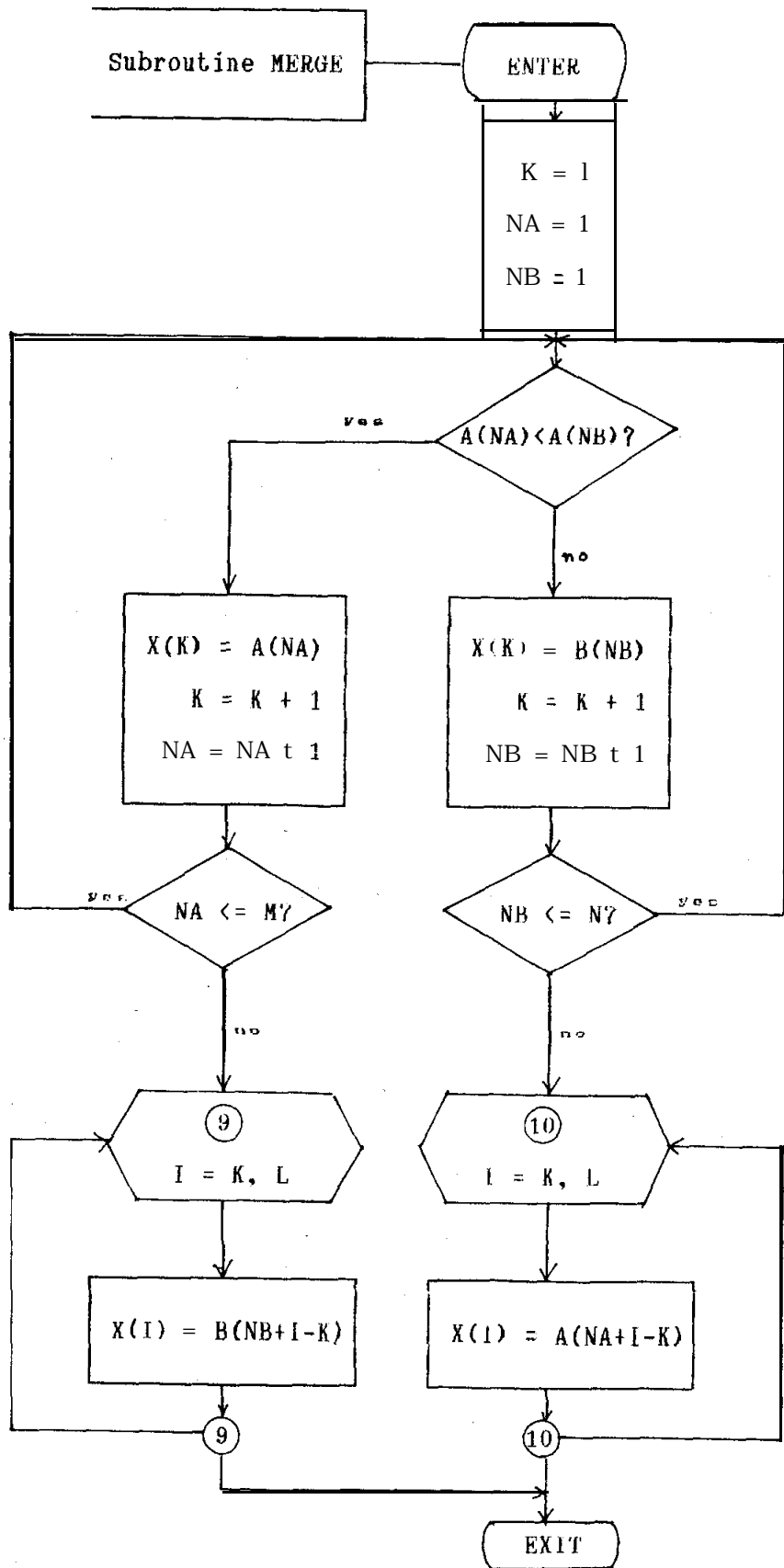
M = ขนาดของอะเรย์ A , NA = 1, 2, ..., M ----> subscript ของ A

N = ขนาดของอะเรย์ B , NB = 1, 2, ..., N ----> subscript ของ B

L = M+N ขนาดของอะเรย์ X , K = 1, 2, ..., L ----> subscript ของ X

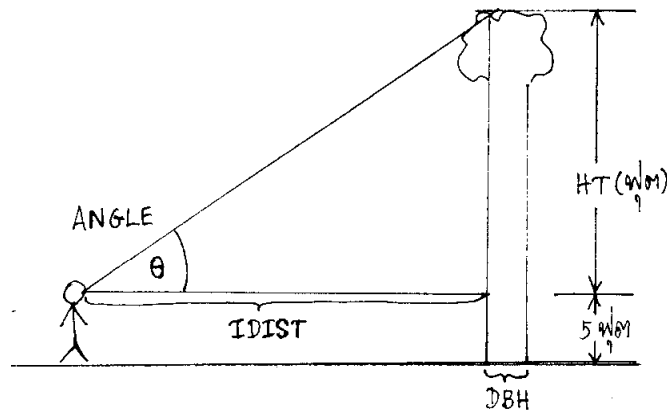
นักศึกษาอาจใช้ SUBROUTINE MERGE(A,M,B,N,X,L)

DIMENSION A(M), B(N), X(L)



20. ข้อมูลจากต้นไม้ประเภทสนต้นหนึ่ง ๆ บันทึกใน 1 เรคคอร์ด ซึ่งมี format ดังต่อไปนี้

column	รายการข้อมูล	variable
1-3	เลขประจำแปลง (plot number)	NPLOT
4-6	เลขประจำต้นไม้ที่อยู่ในแต่ละแปลง (number of tree)	NTREE
7-10	เส้นผ่าศูนย์กลางของต้นไม้ (วัดเป็นนิ้ว มีทศนิยม 1 ตำแหน่ง)	DBH
11-13	ระยะที่คนยืนห่างจากต้นไม้ (วัดเป็นฟุต)	IDIST
14-17	มุมที่มองไปยังยอดไม้ (วัดเป็นองศา มีทศนิยม 1 ตำแหน่ง)	ANGLE



จงเขียนโปรแกรมเพื่อนับจำนวนต้นไม้ในแต่ละแปลง (NO. OF TREES)
คำนวณหาปริมาตรรวมของต้นไม้ในแต่ละแปลง และหาปริมาตรรวม
ของต้นไม้ทั้งหมด (รวมทั้งแปลง) ด้วย

กำหนด output, format.

แต่ละแปลงให้พิมพ์ผลดังนี้

PLOT NUMBER = -----
 NUMBER OF 'TREES (IN THIS PLOT) = -----
 TOTAL CUBIC FEET VOLUMN = -----

ตัวอย่าง แปลงที่ 1 มีต้นไม้ 5 ต้น จะพิมพ์ดังนี้

PLOT NUMBER = 101

NUMBER OF TREES (IN THIS PLOT) = 5

TOTAL VOLUME (CUBIC FEET) = -----

↑
ปริมาตรของต้นไม้ทั้ง 5 ต้นในแปลงนี้

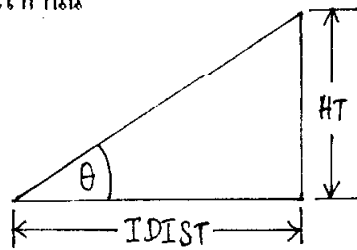
หลังจากพิมพ์รายงานของทุกแปลงแล้วให้พิมพ์ ปริมาตรรวมทั้งหมด

TOTAL VOLUME FOR ALL PLOTS (CUBIC FEET) = -----

สำหรับแปลงที่ไม่มีต้นไม้ให้พิมพ์ข้อความ 2 บรรทัดดังนี้

PLOT NUMBER = -----

NUMBER OF TREE = NONE



กำหนดสูตร

HT (ความสูง วัดเป็นฟุต) = IDIST * tan(θ)

หา tan(θ) โดยใช้ฟังก์ชัน TAN (θ วัดเป็นเรเดียน)

(2π เรเดียน = 360 องศา, π = 3.141593)

ถ้าใช้ Micro Vax II อาจหา tan(θ) โดยใช้ฟังก์ชัน TAND(θ)

(θ วัดเป็นองศา)

สมมติว่าปริมาตรต้นไม้หาได้จากสูตร

ความสูงของคนวัดมุม (วัดถึงระดับตา)

ปริมาตร = 1.6693 + 0.00297 * (DBH)² * (HT + 5.0)

(Volume)

↓
ความสูงของต้นไม้

หมายเหตุ อย่าลืมเปลี่ยน DBH ให้มีหน่วยเป็นฟุตก่อน เพราะให้หาปริมาตรมี

หน่วยเป็น ft³ (CUBIC FT.)

ค่าคงที่ในสูตรหาปริมาตร เป็นตัวปรับค่า เนื่องจากต้นไม้ไม่อยู่ในรูป

ทรงกระบอกที่เดียว

กำหนดข้อมูลเข้า

เขตคอร์ด	คอลัมน์ 123456789012.345678901234567690	หมายเหตุ
1	101 01 100 35 575	ไม่มีต้นไม้ในแปลงนี้
2	101 02 125 41 556	
3	101 03 93 38 528	
4	101 04 132 30 634	
5	101 05 146 45 553	
6	102 00	
7	103 01 80 38 498	
8	103 02 105 39 547	
9	103 03 120 43 544	
10	103 04 131 43 565	
11	103 05 125 40 584	
12	103 06 90 30 614	
13	103 07 Y5 36 568	
14	104 01 102 38 577	
15	104 02 125 42 571	
16	104 03 125 48 536	
17	105 01 132 41 557	
18	105 02 95 39 547	
19	105 03 146 59 609	
20	105 04 155 28 682	
21	106 00	
22	107 01 70 37 472	
23	107 02 65 37 434	

22. จงสร้าง input file เพื่อเก็บข้อมูลเข้า 20 เรคคอร์ด แต่ละเรคคอร์ด
มีค่าของ X และ Y ตามที่กำหนดให้ข้างล่าง

จงเขียนโปรแกรมเพื่อคำนวณหาค่า Z จากแต่ละเรคคอร์ดตามเงื่อนไขดังนี้

ก. ถ้า $X < 1,000$ และ $Y \leq 1,500$ คำนวณ $Z = 2X^{2/3} - Y^{1/3}$

ข. ถ้า $X \geq 1,000$ และ $Y > 1,500$ คำนวณ $Z = X^{1/2} + Y^{2/3}$

นอกจากนี้ไม่ต้องคำนวณค่า Z แต่ให้พิมพ์ค่า X และ Y ด้วย (ค่าของ Z
ให้เว้นว่างไว้)

กำหนด Output format

```

                COMPUTED  Z-VALUES
      ชุดที่ -->NUMBER      X          Y          Z
      -
      :
```

กำหนด Input data

เรคคอร์ดที่	X	Y	เรคคอร์ดที่	X	Y
1	691.21	7012.64	11	1724.72	1477.76
2	676.58	325.86	12	1024.72	1728.00
3	866.79	507.20	13	7195.48	2234.67
4	1525.72	2950.46	14	122.92	322.61
5	3996.42	3248.78	15	856.53	1640.80
6	1477.80	1478.01	16	3115.35	1014.70
7	330.08	1906.96	17	729.00	512.00
8	1618.42	2908.00	18	142.67	209.79
9	4663.40	2002.67	19	1627.50	2707.20
10	4110.23	2226.06	20	437.72	395.60

23. จากข้อ 19.

23.1 ในการสร้างเลขสุ่มตัวหนึ่ง (z_i) ซึ่งเป็นค่า ๆ หนึ่งของตัวแปรเชิงสุ่ม Z ซึ่งมีการแจกแจงแบบปกติมาตรฐาน [Standard Normal Distribution, i.e. $N(0,1)$] จากเลขสุ่มซึ่งมีการแจกแจงแบบยูนิฟอร์มนั้นทำได้โดยใช้ขั้นตอนวิธีดังนี้

1. สร้างเลขสุ่มซึ่งมาจากการแจกแจงแบบยูนิฟอร์ม 12 ตัว (u_1, u_2, \dots, u_{12}) โดยใช้ชื่อบริษัท RANDOM ที่กำหนดไว้ในข้อ 19.)
2. หา $\sum_{i=1}^{12} u_i$ เก็บใน SUM
3. $z_i = \text{SUM} - 6$ จะเป็นค่าหนึ่งของตัวแปรเชิงสุ่ม Z ซึ่งมีการแจกแจงเป็น $N(0,1)$

จงเขียนโปรแกรมเพื่อสร้างค่า $z_i, i=1, \dots, 50$ แล้วคำนวณหา

$$\bar{z} = \sum_{i=1}^{50} z_i / 50 \text{ และ s.d.} = \left\{ (1/49) \left[\sum_{i=1}^{50} z_i^2 - \left(\sum_{i=1}^{50} z_i \right)^2 / 50 \right] \right\}^{(1/2)}$$

กำหนด output format

RANDOM SAMPLE OF SIZE 50

FROM $N(0,1)$

z_1	z_2	z_3	...	z_{10}
z_{11}	z_{12}	z_{13}	...	z_{20}
...				
z_{41}	z_{42}	z_{43}	...	z_{50}

SAMPLE MEAN = _____

SAMPLE S.D. = _____

23.2 ถ้าต้องการ x_i ซึ่งเป็นค่าหนึ่งของตัวแปรเชิงสุ่ม $X \sim N(\mu, \sigma^2)$

เราทำได้โดยใช้สูตร

$$x_i = \mu + \sigma z_i$$

โดยที่ z_i คือค่า ๆ หนึ่งของ $Z \sim N(0, 1)$

ถ้ากำหนดให้ $\mu = 25$ และ $\sigma = 2.5$ จงเขียนโปรแกรมเพื่อสร้างค่า $x_i, i=1, \dots, 50$ [$x_i, i=1, \dots, 50$ คือตัวอย่างสุ่มขนาด 50 จากประชากรซึ่งมีการแจกแจงเป็น $N(25, 2.5^2)$] แล้วคำนวณหา

$$\bar{x} = \sum_{i=1}^{50} x_i / 50 \text{ และ s.d.} = \left\{ (1/49) \left[\sum_{i=1}^{50} x_i^2 - \left(\sum_{i=1}^{50} x_i \right)^2 / 50 \right] \right\}^{(1/2)}$$

กำหนด output format

RANDOM SAMPLE: OF SIZE 50

PROM N(25,6.25)

x_1	x_2	x_3	...	x_{10}
x_{11}	x_{12}	x_{13}	...	x_{20}
			...	
x_{41}	x_{42}	x_{43}	...	x_{50}

SAMPLE MEAN = _____

SAMPLE S.D. = _____

23.3 ถ้าต้องการค่า y ซึ่งเป็นค่า ๆ หนึ่งของตัวแปรเชิงสุ่ม chi-square $Y \sim \chi_m^2$

เราทำได้โดยใช้สูตร

$$y = \sum_{i=1}^m z_i^2$$

โดยที่ z_i คือค่า ๆ หนึ่งของ $Z \sim N(0,1)$

ถ้ากำหนดให้ $m = 15$ จงเขียนโปรแกรมเพื่อสร้างค่า $y_i, i=1, \dots, 100$

แล้ว นับจำนวนตัวของ $y_i < 7.261 = \chi^2_{15, .05}$ แล้วเก็บใน O_1

นับจำนวนตัวของ $7.261 \leq y_i \leq 24.996$ แล้วเก็บใน O_2

นับจำนวนตัวของ $y_i > 24.996 = \chi^2_{15, .05}$ แล้วเก็บใน O_3

กำหนด $E_1 = 5, E_2 = 90,$ และ $E_3 = 5$

ทำการทดสอบ Goodness of fit (อย่างคร่าว ๆ) โดยคำนวณค่า

$$\chi^2_c = \sum_{i=1}^3 [(O_i - E_i)/E_i]^2$$

เทียบกับ $\chi^2_{2, .05} = 5.991$

ก) ถ้า $\chi^2_c \geq 5.991$ ให้พิมพ์สรุปว่า

CONCLUSION: RANDOM NUMBERS **ARE: NOT** FROM CHI-SQUARE DISTRIBUTION
(15 D.F.).

ข) ถ้า $\chi^2_c < 5.991$ ให้พิมพ์สรุปว่า

CONCLUSION: RANDOM NUMBERS ARE FROM CHI-SQUARE DISTRIBUTION
(15 D.F.).

กำหนดให้พิมพ์ผลดังนี้ GOODNESS OF FIT TEST

VALUE	$Y < 7.261$	$7.261 \leq Y \leq 24.996$	$Y > 24.996$	TOTAL
O(I)	ค่า O_1	ค่า O_2	ค่า O_3	100
E(I)	5	90	5	100

CHI-SQUARE (COMPUTED) = _____

CHI-SQUARE (2, .05) = 5.991

CONCLUSION: ก) หรือ ข)

TEST3.LIS

```

0001 C*****
0002 C   PROGRAM      : TEST3 ***** (NEW)*****
0003 C   PROGRAMER   :
0004 C   ID          :
0005 C*****
0006      FUNCTION1(X) = 1/(1+X**2)
0007      OPEN (13,FILE='TEST3.OUT',STATUS='NEW')
0008      DATA N,A,B/100,0.0,1.0/,HL1,HL2/2*0.0/
0009      H=(B-A)/N
0010      D=A-H
0011      DO 100 I=1,N-1,2
0012          D=D+2*H
0013 L00      HL1=HL1+FUNCTION1(D)
0014          HL1=HL1*4
0015          D=A
0016      00 20" ?=2,N-2,2
0017          D=D+2*H
0018 200      HL2=HL2+FUNCTION1(D)
0019          HL2=HL2*2
0020      AREA = (H/3)*(FUNCTION1(A)+FUNCTION1(B)+HL1+HL2)
0021      WRITE (13,10) AREA
0022 10      FORMAT (5X,'AREA UNDER CURVE Y=1/(1+X**2)X FROM 0.0 TO 1.0',/,
0023          # 5X,'BY SIMPSON'S RULE : AREA = ',F10.7)
0024      CLOSE (13,STATUS='KEEP')
0025      STOP
0026      END

```

PROGRAM SECTIONS

NAME	BITES	ATTRIBUTES
0 \$CODE	200	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	97	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	52	PIC CON REL LCL NOSHR NOEXE RO WRT LONG
TOTAL SPACE ALLOCATED	349	

ENTRY POINTS

ADDRESS	TYPE	NAME
0-00000000		TEST3\$MAIN

STATEMENT FUNCTIONS

ADDRESS	TYPE	NAME
**	R**	FUNCTION1

TEST3.LIS (M0)

TEST3\$MAIN

VARIABLES

ADDRESS	TYPE	NAME	ADDRESS	TYPE	NAME	ADDRESS	TYPE	NAME	ADDRESS	TYPE	NAME
**	R*4	A	**	R*4	AREA	**	R*4	B	**	R*4	D
**	K*4	H	2-00000000	R*4	HL1	2-00000004	R*4	PL2	**	T*4	I
**	I*4	N	AP-00000004	R*4	X						

LABELS

ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL
1-0000000A	10'	**	100	**	200

FUNCTIONS AND SUBROUTINES REFERENCED

TYPE	NAME	TYPE	NAME
	FOR\$CLOSE		FOR\$OPEN

COMMAND QUALIFIERS

FOR TEST3/LIS

```

/CHECK=(NOBCOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS)
/CONTINUATIONS=19 /NOCPDSS_REFERENCE /NOD_LINES /NOEXTEND_SOURCE /F77
/NOG_FLOATING /i4 /NOMACHINE_CODE /OPTIMIZE

```

COMPILATION STATISTICS

```

RUN TIME:          2.11 SECONDS
ELAPSED TIME:     3.12 SECONDS
PAGE FAULTS:      534
DYNAMIC MEMORY:   335 PAGES

```

TEST3.OUT

```

AREA UNDER CURVE Y=1/(1+X**2) X FROM 0.0 TO 1.0
BY SIMPSON'S RULE : AREA = 0.7853981

```

3. ตัวอย่างการวิ่งโปรแกรมด้วยเครื่องไมโครคอมพิวเตอร์

(Microsoft FORTRAN77 V3.31)

ตัวอย่างที่ 1

โปรแกรมที่มี compile-time errors (ในแฟ้ม TRY.FOR)

C234567890

C PROGRAM TO SHDW COMPILATION ERRORS

C

OPEN(5,FILE='A:TRY.DAT',STATUS='OLD')

OPEN(6,FILE='A:TRY.OUT',STATUS='NEW')

DIMENSION X(30)

WRITE(6,10)

10 FORMAT(5X,'I',10X,'X(I)')

READ(5,2)X(I),I=1,30

12 **FORMAT(L6F5.2)**

WRITE(6,99)(I,X(I),I=1,30)

99 FORMAT(1X,I4,2X,F8.2

XMAX=X(1)

XMIN=X(1)

DO 13 I=2,30

IF(X(I) .LE. XMAX)GO TO 14

XMAX=X(I)

IMAX+2=I

14 IF(X(I) .GE. XMIN)GO TO 13

XMIN=X(I)

IMIN=I

13 **CONTINUE**

WRITE(6, 15)XMAX,IMAX,XMIN,IMIN

15 FORMAT(5X,'MAXIMUM VALUE IS',F8.2,' ITS POSITION IS',I3/

t 5X,'MINIMUM VALUE IS',F8.2,' ITS POSITION IS',I3)

END

STOP

```

D Line# 1      7      Microsoft  FORTRAN 77  V3.31  August  1985
  1 C234567890
  2 C      PROGRAM TO SHOW COMPILATION ERRORS
  3 c
  4      OPEN(5,FILE='A:TRY.DAT',STATUS='OLD')
  5      OPEN(6,FILE='A:TRY.OUT',STATUS='NEW')
  6      DIMENSION X(30)
**** Error 100 -- statement order
  7      WRITE(6,10)
  8 10     FORMAT(5X,'I',10X,'X(I)')
  9      READ(5,2)X(I),I=1,30
**** Error 56 -- too many subscripts
**** Error 162 -- unrecognizable I/O list element
**** Error 34 -- identifier already declared
**** Error 34 -- identifier already declared
**** Error 34 -- identifier already declared
 10 12     FORMAT(16F5.2)
 11      WRITE(6,99)(I,X(I),I=1,30)
**** Error 56 -- too many subscripts
 12 99     FORMAT(1X,I4,2X,F8.2)
**** Warning -- Invalid format, Error 1200
 13      XMAX=X(1)
**** Error 56 -- too many subscripts
 14      XMIN=X(1)
**** Error 56 -- too many subscripts
 15      00 13 I=2,30
 1  16     IF(X(I) .LE. XMAX)GO TO 14
**** Error 56 -- too many subscripts
 1  17     XMAX=X(I)
**** Error 56 -- too many subscripts
 1  18     IMAX+2=I
**** Error 89 -- unrecognizable statement
 1  19 14   IF(X(I) .GE. XMN)GO TO 13
**** Error 56 -- too many subscripts
 1  20     XMN=X(I)
**** Error 56 -- too many subscripts
 1  21     IMN=I
 1  22 13   CONTINUE
 23      WRITE(6,15)XMAX,IMAX,XMIN,IMIN
 24 15     FORMAT(5X,'MAXIMUM VALUE IS',F8.2,' ITS POSITION IS',I3/
 25      t      5X,'MINIMUM VALUE IS',F8.2,' ITS POSITION IS',I3)
 26      END
Format 2      not defined.
**** Error 163 -- FORMAT not found

```


0 Line# 1 7 Microsoft FORTRAN77 V3.31 August 1985

```

1 C234567890
2 C
3     DIMENSION X(30)
4     OPEN(5,FILE='A:TRY.DAT',STATUS='OLD')
5     OPEN(6,FILE='A:TRY.OUT',STATUS='NEW')
6     WRITE(6,10)
7 10  FORMAT(1X,'30 OBSERVATIONS'/4X,'I',6X,'X(I)'/)
8     READ(5,2)(X(I),I=1,30)
9 2   FORHAT(16F5.2)
10    WRITE(6,99)(I,X(I),I=1,30)
11 99  FORMAT(1X,I4,2X,F8.2)
12    XMAX=X(1)
13    IMAX=1
14    XMIN=X(1)
15    IMIN=1
16    DO 13 I=2,30
1 17    IF(X(I) .LE. XMAX)GO TO 14
1 18    XMAX=X(I)
1. 19    IMAX=I
1 20 14 IF(X(I) .GE. XMN)GO TO 13
1 21    XMIN=X(I)
1 22    IMIN=I
1 23 13 CONTINUE
24    WRITE(6,15)XMAX,IMAX,XMIN,IMIN
25 15  FORMAT(5X,'MAXIMUM VALUE IS',F8.2,' ITS POSITION IS',I3/
26 t   5X,'MINIMUM VALUE IS',F8.2,' ITS POSITION IS',I3)
27    STOP
28    END

```

Name	Type	Offset	P	Class
I	INTEGER*4	176		
IMAX	INTEGER*4	212		
IMN	INTEGER*4	220		
X	REAL	16		
XMAX	REAL	208		
XMN	REAL	216		

Name	Type	Size	Class
MAIN			PROGRAM

Pass One No Errors Detected
28 Source Lines

TRY.DAT

10.98 11.13 12.51 8.4 9.27 8.73 6.36 8.5 8.82 9.14 35.3 29.7 30.8 58.8 61.4 71.3
74.4 70.7 57.5 44.4 7.82 8.24 12.19 11.88 9.57 10.94 9.58 10.09 8.11 6.83

TRY.OUT

30 OBSERVATIONS

I	X(I)
1	10.98
2	11.13
3	12.51
4	8.40
5	9.27
6	8.73
7	6.36
8	8.50
9	8.82
10	9.14
11	5.30
12	9.70
13	.81
14	8.81
15	1.47
16	.30
17	74.40
18	70.70
19	57.50
20	46.40
21	7.82
22	8.24
23	2.19
24	1.88
25	9.57
26	.94
27	9.58
28	.09
29	8.11
30	6.83

MAXIMUM VALUE IS 74.40 **ITS POSITION IS** 17
MINIMUM VALUE IS .09 **ITS POSITION IS** 28

ตัวอย่างที่ 2 NEWTON-FOR

```
C234567890
C      NEWTON'S METHOD
C      FOR FINDING ROOTS OF A QUADRATIC FUNCTION
C
      OPEN(6,FILE='A:NEWTON.OUT',STATUS='NEW')
      WRITE(6,10)
10     FORMAT(5X,'ROOTS OF A FUNCTION' /
+5X,'F(X)=X**2+3.4*X-6.11'/)
C
C      THE FIRST GUESS FOR THE FIRST ROOT
C
      x=25
      DO 19 I=1,2
      WRITE(6,11)
11     FORMAT(7X,'X',10X,'F(X)')
17     Y=X**2+3.4*X-6.11
      WRITE(6,12)X,Y
12     FORMAT(2F12.5)
      IF(Y .LE. .00001)GO TO 16
      Z=2.0*X+3.4
      X=X-Y/Z
      GO TO 17
16     WRITE(6,22)I,X
22     FORMAT(10X,'ROOT(',I1,')=' ,F5.1/)
C
C      THE SECOND GUESS FOR THE SECOND ROOT
C
      X=-10
19     CONTINUE
      STOP
      END
```

D Line#C234567890

Microsoft FORTRAN77 V3.31 August 1985

```

2 C      NEWTON'S METHOD
3 c      FOR FINDING ROOTS OF A QUADRATIC FUNCTION
4 c
5      OPEN(6,FILE='A:NEWTON.OUT',STATUS='NEW')
6      WRITE(6,10)
7 10     FORMAT(5X,'ROOTS OF A FUNCTION' /
8      +5X,'F(X)=X**2+3.4*X-6.11' /)
9 c
10 c     THE FIRST GUESS FOR THE FIRST ROOT
11 c
12      x=25
13      DO 19 I=1,2
1 14     WRITE(6,11)
1 15 11   FORMAT(7X,'X',10X,'F(X)')
1 16 17   Y=X**2+3.4*X-6.11
1 17     WRITE(6,12)X,Y
1 18 12   FORMAT(2F12.5)
1 19     IF(Y .LE. .00001)GO TO 16
1 20     Z=2.0*X+3.4
1 21     x=x-v/z
1 22     GO TO 17
1 23 16   WRITE(6,22)I,X
1 24 22   FORMAT(10X,'ROOT(',I1,')=',F5.1/)
1 25 C
1 26 c     THE SECOND GUESS FOR THE SECOND ROOT
1 27 C
1 28     X=-10
1 29 19   CONTINUE
30     STOP
31     END

```

Name	Type	Offset	P	Class
I	INTEGER*4	152		
X	REAL	148		
Y	REAL	176		
Z	REAL	188		

Name	Type	Size	Class
MAIN			PROGRAM

Pass One No Errors Detected
31 Source Lines

NEWTON.OUT

ROOTS OF A FUNCTION
 $F(X)=X**2+3.4*X-6.11$

X	F(X)
25.00000	703.89000
11.81854	173.75090
5.39215	41.29854
2.48058	8.47723
1.46670	1.02796
1.30439	.02634
1.30000	.00002
1.30000	.00000

ROOT(1) = 1.3

X	F(X)
-10.00000	59.89000
-6.39217	13.01645
-5.00513	1.92388
-4.71409	.08471
-4.70003	.00020
-4.70000	.00000

ROOT(2) = -4.7

ตัวอย่างที่ 3 EXTRA.FOR

```
C234567890
    DIMENSION Y(20),Z(30),X(50)
    OPEN(5,FILE='A:EXTRA.DAT',STATUS='OLD')
    OPEN(6,FILE='A:EXTRA.OUT',STATUS='NEW')
C    READ 20 Y-VALUES FROM 4 RECORDS
C    READ 30 Z-VALUES FROM THE NEXT 6 RECORDS
    READ(5,10)Y,Z
10   FORMAT(5F5.2)
C    REWIND TO THE BEGINNING OF FILE EXTRA.DAT
    REWIND(5)
C    REAO 50 X-VALUES FROM 10 RECORDS
    READ(5,10)X
    WRITE(6,12)Y
12   FORMAT(1X,'Y-VALUES:',10F6.2/10X,10F6.2)
    CALL STAT(Y,20)
    WRITE(6,13)Z
13   FORMAT(1X,'Z-VALUES:',10F6.2/(10X,10F6.2))
    CALL STAT(Z,30)
    WRITE(6,14)X
14   FORMAT(1X,'X-VALUES:',10F6.2/(10X,10F6.2))
    CALL STAT(X,50)
    STOP
    END
```

```
C234567890
    SUBROUTINE STAT(A,N)
    DIMENSION A(N)
c    INITIALIZE SUMA AND SUMAA
C    YOU MUST NOT USE DATA STATEMENT TO DO THIS
    SUMA=0
    SUMAA=0
    DO 10 I=1,N
    SUMA=SUMA+A(I)
10   SUMAA=SUMAA+A(I)*A(I)
    ABAR=SUMA/N
    SDA=SQRT((SUMAA-SUMA*SUMA/N)/(N-1.0))
    WRITE(6,5)SUMA,SUMAA,ABAR,SDA
5    FORMAT(/1X,'SUM      =',F12.4/
t      1X,'SUM-SQ  =',F12.4/
t      1X,'MEAN   =',F12.4/
t      1X,'S.D.   =',F12.4/)
    RETURN
    END
```

D Line# 1 7 Microsoft FORTRAN77 V3.31 August 1985

```

1 C234567890
2   DIMENSION Y(20),Z(30),X(50)
3   OPEN(5,FILE='A:EXTRA.DAT',STATUS='OLD')
4   OPEN(6,FILE='A:EXTRA.OUT',STATUS='NEW')
5 c   READ 20 Y-VALUES FROM 4 RECORDS
6 C   READ 30 Z-VALUES FROM THE NEXT 6 RECOROS
7     READ(5,10)Y,Z
8 10  FORMAT(5F5.2)
9 c   REWIND TO THE BEGINNING OF FILE EXTRA.DAT
10  REWIND(S)
1.1 c READ 50 X-VALUES FROM 10 RECORDS
12  READ(5,10)X
13  WRITE(6,12)Y
14 12  FORMAT(1X,'Y-VALUES:',10F6.2/10X,10F6.2)
15  CALL STAT(Y,20)
16  WRITE(6,13)Z
17 13  FORMAT(1X,'Z-VALUES:',10F6.2/(10X,10F6.2))
18  CALL STAT(Z,30)
19  WRITE(6,14)X
20 14  FORMAT(1X,'X-VALUES:',10F6.2/(10X,10F6.2))
21  CALL STAT(X,50)
22  STOP
23  END

```

Name	Type	Offset	P	Class
X	REAL	216		
Y	REAL	16		
Z	REAL	96		

```

24 C234567890
25   SUBROUTINE STAT(A,N)
26   DIMENSION A(N)
27 C   INITIALIZE SUMA AND SUMAA
28 C   YOU MUST NOT USE DATA STATEMENT TO DO THIS
29   SUMA=0
30   SUMAA=0
31   DO 10 I=1,N
1 32   SUMA=SUMA+A(I)
1 33 10  SUMAA=SUMAA+A(I)*A(I)
34   ABAR=SUMA/N
35   SDA=SQRT((SUMAA-SUMA*SUMA/N)/(N-1.0))
36   WRITE(6,5)SUMA,SUMAA,ABAR,SDA
37 5   FORMAT(/1X,'SUM   =',F12.4/
38 t   1X,'SUM-SQ  =',F12.4/
39 t   1X,'MEAN   =',F12.4/
40 t   1X,'S.D.   =',F12.4/)
41   RETURN
42   END

```

Name	Type	Offset	P	Class
A	REAL	0	*	
ABAR	REAL	600		
I	INTEGER*4	592		

Page 2
04-23-82
18:06:35

D Line# 1 7
N INTEGER*4 4 *
SDA REAL 604
SPRT INTRINSIC
SUMA REAL 584
SUMAA REAL 588

Microsoft FORTRAN77 V3.31 August 1985

Name	Type	Size	Class
MAIN			PROGRAM
STAT			SUBROUTINE

Pass One No Errors Detected
42 Source Lines

EXTRA. DAT

L0. 9811. 1312. 51 8. 4 9. 27
8. 73 6. 36 8. 5 8. 82 9. 14
35. 3 29. 7 30. 8 58. 8 61. 4
71. 3 74. 4 70. 7 57. 5 46. 4
7. 82 8. 2412. 1911. 88 9. 57
10. 94 9. 5810. 09 8. 11 6. 83
28. 9 28. 1 29. 1 46. 8 48. 5
59. 3 70. 0 70. 0 74. 5 71. 3
8. 88 7. 68 8. 47 8. 8610. 36
11. 0813. 4510. 0012. 12 9. 2
72. 1 58. 1 44. 6 33. 4 28. 6
32. 7 57. 4 49. 7 34. 6 60. 9

EXTRA. OUT

Y-VALUES: 10. 98 11. 13 12. 51 8. 40 9. 27 8. 73 6. 36 8. 50 8. 82 9. 14
35. 30 29. 70 30. 80 58. 80 61. 40 71. 30 74. 40 70. 70 57. 50 46. 40

SUM = 630.1400
SUM W = 32288. 6500
MEAN = 31. 5070
S. D. = 25. 5825

Z-VALUES: 7. 82 8. 24 12. 19 11. 88 9. 57 10. 94 9. 58 10. 09 8. 11 6. 83
28. 90 28. 10 29. 10 46. 80 48. 50 59. 30 70. 00 70. 00 74. 50 71. 30
8. 88 7. 68 8. 47 8. 86 10. 36 11. 08 13. 45 10. 00 12. 12 9. 20

SUM = 721. 8500
SUM-SQ = 32931. 2100
MEAN = 24. 0617
S. D. = 23. 1653

X-VALUES: 10. 98 11. 13 12. 51 8. 40 9. 27 8. 73 6. 36 8. 50 8. 82 9. 14
35. 30 29. 70 30. 80 58. 80 61. 40 71. 30 74. 40 70. 70 57. 50 46. 40
7. 82 8. 24 12. 19 11. 88 9. 57 10. 94 9. 58 10. 09 8. 11 6. 83
28. 90 28. 10 29. 10 46. 80 48. 50 59. 30 70. 00 70. 00 74. 50 71. 30
8. 88 7. 68 8. 47 8. 86 10. 36 11. 08 13. 45 10. 00 12. 12 9. 20

SUM = 1351. 9900
SUM W = 65219. 8600
MEAN = 27. 0398
S. D. = 24. 1856

Appendix E

Diagnostic Messages

Diagnostic messages related to a VAX FORTRAN program can come from the compiler, the linker, or the VAX run-time system. The compiler detects syntax errors in the source program, such as unmatched parentheses, invalid characters, misspelled keywords, and missing or invalid parameters. The run-time system reports errors that occur during execution.

This chapter lists and describes the messages issued by the compiler and the run-time system. It also provides a summary of the DICTONARY messages that may accompany Common Data Dictionary messages. Linker messages are summarized in the *VAX/VMS Linker Reference Manual*.

E.1 Diagnostic Messages **from** the Compiler

A diagnostic message issued by the compiler describes the detected error, and in some cases contains an indication of the action taken by the compiler in response to the error.

Besides reporting errors detected in source program syntax, the compiler issues messages indicating errors that involve the compiler itself, such as I/O errors.

E.1.1 Source Program Diagnostic Messages

There are four classes of source program diagnostic messages. In order of greatest to least severity, these classes are:

Code	Description
F	Fatal; must be corrected before the program can be compiled. No object file is produced if an F-class error is detected during compilation.
E	Error; should be corrected. An object file is produced despite the E-class error, but the output or program result may be incorrect.

Code	Description
W	Warning; should be investigated by checking the statements to which W-class diagnostic messages apply. Warnings are issued for statements that use acceptable, but nonstandard, syntax and for statements corrected by the compiler. An object file is produced, but the program results may be <i>incorrect</i> . Note that W-class messages are produced unless the /NOWARNINGS qualifier is specified in the FORTRAN command.
I	Information; not an <i>error</i> message and does not call for corrective action. However, the I-class message informs you that either a correct VAX FORTRAN statement may have unexpected results or you have used a VAX extension to FORTRAN-77.

Typing mistakes are a likely cause of syntax errors: they can cause the compiler to generate misleading diagnostic messages. Beware especially of the following:

- Missing comma or parenthesis in a complicated expression or FORMAT statement.
- Misspelled variable names. The compiler may not detect this error, so execution can be affected.
- Inadvertent line continuation mark. This can cause a diagnostic message for the preceding line.
- Extension of the statement line past column 72. Unless /EXTEND-SOURCE is specified, this can cause diagnostic messages because the statement is terminated early.
- Confusion between the digit 0 and the uppercase letter O. This can result in variable names that appear identical to you but not to the compiler.

Another source of diagnostic messages is the inclusion of invalid ASCII characters in the source program. With the exception of the tab, space, and form-feed characters, nonprinting ASCII control characters are not valid in a FORTRAN source program. As the source program is scanned, such invalid characters are replaced by a question mark (?). However, because the question mark cannot occur in a FORTRAN statement, a syntax error usually results.

Because a diagnostic message indicates only the immediate cause, you should always check the entire source statement carefully.

The following examples show how source program diagnostic messages are displayed in interactive mode at your terminal. Figure E-1 shows how these messages appear in listings.

```

%FORT-W-FMTEXTCOM, Extra comma in format list
      [FORMAT (I3,)] in module MORTGAGE at line 13

%FORT-F-UNDSTALAB, Undefined statement label
      [66] in module MORTGAGE at line 19

%FORT-F-ENDNOOBJ, DB1:CSMITH1MOR.FOR:1 completed
with 2 diagnostics - object deleted

0001      C      Program to calculate monthly mortgage payments
0002
0003          PROGRAM MORTGAGE
0004
0005          TYPE 10
0006      10      FORMAT (' ENTER AMOUNT OF MORTGAGE ')
0007          ACCEPT 20, IPV
0008      20      FORMAT (I6)
0009
0010          TYPE 30
0011      30      FORMAT (' ENTER LENGTH OF MORTGAGE IN MONTHS ')
0012          ACCEPT 40, IMON
0013      40      FORMAT (I3,)
0014
%FORT-W-FMTEXTCOM, Extra comma in format list
      [FORMAT (I3,)] in module MORTGAGE at line 13

0015          TYPE 50
0016      50      FORMAT (' ENTER ANNUAL INTEREST RATE ')
0017          ACCEPT 60, YINT
0018      60      FORMAT (F6,4)
0019          GO TO 66
0020      65      YI = YINT/12      !Get monthly rate
0021          IMON = -IMON
0022          FIPV = IPV * YI
0023          YI = YI + 1
0024          FIMON = YI**IMON
0025          FIMON = 1 - FIMON
0026          FMNTHLY = FIPV/FIMON
0027
0028          TYPE 70, FMNTHLY
0029      70      FORMAT (' MONTHLY PAYMENT EQUALS ', F7,3)
0030          STOP
0031          END
%FORT-F-UNDSTALAB, Undefined statement label
      [66] in module MORTGAGE at line 19

```

Figure E-1: Sample Diagnostic Messages (Listing Format)

Table E-1 is an alphabetical list of FORTRAN diagnostic error messages. For each message, the table gives a mnemonic, an error code level, the text of the message, and an explanation of the message.

Table E-1: Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
ADJARRBOU	E	Adjustable array bounds must be dummy arguments or in common Variables specified in dimension declarator expressions must either be subprogram dummy arguments or appear in common.
ADJARRUSE	F	Adjustable array used in invalid context A reference was made to an adjustable array in a context where such a reference is not allowed.
ADJLENUSE	F	Passed-length character name used in invalid context A reference was made to a passed-length character array or variable in a context where such reference is not allowed.
ALTRETLAB	F	Alternate return label used in invalid context An alternate return argument was used in a function reference.
ALTRETOMI	E	Alternate return omitted in SUBROUTINE or ENTRY statement An asterisk is missing in the argument list of a subroutine for which an alternate return is specified. Examples: <pre> 1 . SUBROUTINE XYZ(A,B) RETURN 1 2 . ENTRY ABC(Q,R) RETURN 1 </pre>

E-4 Diagnostic Messages

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error t i d e	Text/Meaning
ALTRETSPE	F	<p>Alternate return specifier invalid in FUNCTION subprogram</p> <p>The argument list of a FUNCTION declaration contains an asterisk, or a RETURN statement in a function subprogram specifies an alternate return. Examples:</p> <ol style="list-style-type: none"> 1. INTEGER FUNCTION TCB(ARG ,* ,X) 2. FUNCTION IMAX <p align="center">RETURN I+J END</p>
ARIVALREQ	F	<p>Character expression where arithmetic value required</p> <p>An expression that must be arithmetic (INTEGER, REAL, LOGICAL, or COMPLEX) was of type CHARACTER.</p>
ASSARRUSE	F	<p>Assumed size array name used in invalid context</p> <p>An assumed size array name was used where the size of the array was also required, for example, in an I/O list.</p>
ASSDOVAR	W	<p>Assignment to DO variable within loop</p> <p>The control variable of a DO loop has been altered within the range of the DO statement.</p>
BADEND	F	<p>END [STRUCTURE UNION MAP] must match top.</p> <p>A STRUCTURE, UNION, or MAP statement did not have a corresponding END STRUCTURE, END UNION, or END MAP statement, respectively.</p>
BADFIELD	F	<p>Field name not defined for this structure.</p> <p>A field name not defined in a structure was used in a qualified reference.</p>
BADRECFEF	F	<p>Aggregate reference where scalar reference required</p> <p>An aggregate reference was used where a scalar reference was required.</p>
CDDBITSIZ	F	<p>CDD field specifies a hit size or alignment. Size or address rounded up to byte alignment.</p> <p>CDD's bit datatype and bit alignment are not supported by FORTRAN.</p>

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
CDDERROR	I	CDD description extraction condition The FORTRAN compiler is in the process of extracting a data definition from the Common Data Dictionary. See the accompanying messages for more information.
CDDNOTSTR	F	CDD record is not a structure CDD record description was not structured. VAX FORTRAN requires structure definitions (elementary field descriptions in CDDL).
CDDRECDIM	F	CDD record is dimensioned VAX FORTRAN does not support dimensioned structures, for example, arrays of structures.
CDDSCALED	W	CDD description specifies a scaled data type VAX FORTRAN does not support scaled data types. The data described by the CDD specifies a scaled component.
CDD'TOOBIG	E	Attributes for some member of CDD record description exceed implementation's limit for member complexity Some member of the CDD record description has too many attributes and has created a program that is too large. Change the Common Data Dictionary description to make the field description smaller.
CDDTOODEEP	E	Attributes for CDD record description exceed implementation's limit for record complexity The CDD record description contains structures that are nested too deeply. Modify the CDD description to reduce the level of nesting in the record description.
CHANAMINC	E	Character name incorrectly initialized with numeric value Character data with a length greater than one was initialized with a numeric value in a data statement. Example: <pre>CHARACTER*4 A DATA A/14/</pre>
CHASBSLIM	F	Character substring limits out of order The first character position of a substring expression is greater than the last character position. Example: <pre>C(5:3)</pre>

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
CHAVALREQ	F	Arithmetic expression where character value required An expression that must be of type CHARACTER was of another data type.
COLMAJOR	F	CDD description specifies that it is not a column major array FORTRAN only supports column-major arrays. Change the CDD description to specify a column-major array.
CONSIZEXC	E	Constant size exceeds variable size in data initialization A constant used for data initialization is larger than its corresponding variable.
DBGOPT	I	The NOOPTIMIZE qualifier is recommended with the DEBUG qualifier. Optimizations performed by the compiler can cause several different kinds of unexpected behavior when using VAX DEBUG. See Chapter 1 of <i>VAX FORTRAN User's Guide</i> for more information.
DEFSTAUNK	I	Default STATUS= 'UNKNOWN ' used in OPEN statement The OPEN statement default STATUS='UNKNOWN' may cause an old file to be inadvertently modified.
DEPENDITEM	I	CDD description contains Depends Item attribute (ignored). FORTRAN does not support the CDD Depends Item attribute. No action is required.
DICTABORI	F	DICTIONARY processing of CDD record description aborted The FORTRAN compiler is unable to process the CDD record description. See the accompanying messages for further information.
ENTDUMVAR	F	ENTRY dummy variable previously used in executable statement The dummy arguments of an ENTRY statement must not have been used previously in an executable statement in the same program unit.
EQVEXPCOM	F	EQUIVALENCE statement incorrectly expands a common block A common block cannot be extended beyond its beginning by an EQUIVALENCE statement.
EXCCHATRU	E	Non-blank characters truncated in string constant A character or Hollerith constant was converted to a data type which was not large enough to contain all the significant characters.

Diagnostic Messages **E-7**

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
EXCDIGTRU	E	Non-zero digits truncated in hex or octal constant An octal or hexadecimal constant was converted to a data type which was not large enough to contain all the significant digits.
EXCNAMDA'I	E	Number of names exceeds number of values in data initialization The number of constants specified in a DATA statement must match the number of variables or array elements to be initialized. The remaining variables andr array elements are not initialized.
EXCVALDAT	E	Number of values exceeds number of names in data initialization The number of variables or array elements to be initialized must match the number of constants specified in data initialization. The remaining constant values are ignored.
EXPSTAOVE	F	Compiler expression stack overflow An expression is too complex or there are too many actual arguments in a subprogram reference. A maximum of 255 actual arguments can be compiled. You can subdivide a complex expression or reduce the number of arguments.
EXTCHAFOL	E	Extra characters following a valid statement Superfluous text was found at the end of a syntactically correct statement. Check for typing or syntax errors.
EXTMIXCOM	I	Extension to FORTRAN-77: Mixed numeric and character elements in common Numeric and character variable and array elements cannot be equivalenced to each other.
EXTMIXEQV	I	Extension to FORTRAN-77: Mixed numeric and character elements in EQUIVALENCE A common block must not contain both numeric and character data.
EXTRECUSE	I	Extension to FORTRAN-77: Nonstandard use of field reference. A record reference (for example, record-nums.field-name) was used in a program compiled with the /STANDARD=[SYNTAX__ALL] qualifier in the FORTRAN command.
EXT__COM	I	Extension to FORTRAN-77: nonstandard comment FORTRAN-77 allows only the characters "C" and "*" to begin a comment line; "D", "d", and "!" are extensions to FORTRAN-77.

E-t? Diagnostic Messages

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
E X T - C O N S T I		<p>Extension to FORTRAN-77: nonstandard constant</p> <p>The following constant forms are extensions to FORTRAN-77:</p> <ul style="list-style-type: none"> • Hollerith nH..... • Typeless 'xxxx 'X or '0000 '0 • Octal "0000 or 00000 • Hexadecimal Zxxxx • Radix-50 nR..... • Complex with PARAMETER components • COMPLEX*16 (www.xxxDn, yyy.zzzDn) • REAL*16 yyy.zzzQn
EXT-FMT	I	<p>Extension to FORTRAN-77: nonstandard FORMAT statement item</p> <p>The following format field descriptors are extensions to FORTRAN-77:</p> <ul style="list-style-type: none"> • \$,O,Z All forms • A,L,I,F,E,G,D Default field width forms • P Without scale factor
EXT-KEY	I	<p>Extension to FORTRAN-77: nonstandard keyword</p> <p>A nonstandard keyword was used.</p>
EXT__LEX	I	<p>Extension to FORTRAN-77: nonstandard lexical item</p> <p>One of the following nonstandard lexical items was used:</p> <ul style="list-style-type: none"> • An alternate return specifier with an ampersand (&) in a CALL statement • The apostrophe (') form of record specifier in a direct access I/O statement • A variable format expression
EXT-NAME	I	<p>Extension to FORTRAN-77: nonstandard name</p> <p>A name longer than six characters or one that contained a dollar sign (\$) or an underscore (_) was used.</p>

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
EXT__OPER	I	<p>Extension to FORTRAN-77: nonstandard operator</p> <p>The operators .XOR., %VAL, %REF, %DESCR, and %LOC are extensions to FORTRAN-77. The standard form of .XOR. is .NEQV. The % operators are extensions provided to allow access to non-FORTRAN parts of the VAX-11 environment.</p>
EXT__SOURC	I	<p>Extension to FORTRAN-77: tab indentation or lowercase source</p> <p>The use of tab indentation or lowercase letters in source code is an extension to FORTRAN-77.</p>
EXT__STMT	I	<p>Extension to FORTRAN-77: nonstandard statement type</p> <p>A nonstandard statement type was used</p>
EXT-SYN	I	<p>Extension to FORTRAN-77: nonstandard syntax</p> <p>One of the following syntax extensions was specified:</p> <ul style="list-style-type: none"> • PARAMETER name = value <ul style="list-style-type: none"> No parentheses • type name/value/ <ul style="list-style-type: none"> Data initialization in type declaration • DATA (ch(exp:exp),v=e2)/values/ <ul style="list-style-type: none"> Substring initialization with implied-DO in DATA statement • CALL name(arg2,,arg3) <ul style="list-style-type: none"> Null actual argument • READ (...),iolist <ul style="list-style-type: none"> Comma between I/O control and element lists • PARAMETER (name2=ABS(name1)) <ul style="list-style-type: none"> Function use in PARAMETER • el ** -e2 <ul style="list-style-type: none"> Two consecutive operators

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning																										
EXT-TYPE	I	<p>Extension to FORTRAN-77; nonstandard date type specification</p> <p>The following DATA type specifications are extensions to FORTRAN-77. The FORTRAN-77 equivalent is given where available. This message is issued when these types are used in the IMPLICIT statement or in a numeric type statement.</p> <table border="0"> <thead> <tr> <th>Extension</th> <th>Standard</th> </tr> </thead> <tbody> <tr> <td>BYTE</td> <td></td> </tr> <tr> <td>LOGICAL*1</td> <td></td> </tr> <tr> <td>LOGICAL*2</td> <td>LOGICAL (with /NOI4 specified only)</td> </tr> <tr> <td>LOGICAL*4</td> <td>LOGICAL</td> </tr> <tr> <td>INTEGER*2</td> <td>INTEGER (with /NOI4 specified only)</td> </tr> <tr> <td>INTEGER*4</td> <td>INTEGER</td> </tr> <tr> <td>REAL*4</td> <td>REAL</td> </tr> <tr> <td>REAL*8</td> <td>DOUBLE PRECISION</td> </tr> <tr> <td>REAL*16</td> <td></td> </tr> <tr> <td>COMPLEX*8</td> <td>COMPLEX</td> </tr> <tr> <td>COMPLEX*16</td> <td></td> </tr> <tr> <td>DOUBLE COMPLEX</td> <td></td> </tr> </tbody> </table>	Extension	Standard	BYTE		LOGICAL*1		LOGICAL*2	LOGICAL (with /NOI4 specified only)	LOGICAL*4	LOGICAL	INTEGER*2	INTEGER (with /NOI4 specified only)	INTEGER*4	INTEGER	REAL*4	REAL	REAL*8	DOUBLE PRECISION	REAL*16		COMPLEX*8	COMPLEX	COMPLEX*16		DOUBLE COMPLEX	
Extension	Standard																											
BYTE																												
LOGICAL*1																												
LOGICAL*2	LOGICAL (with /NOI4 specified only)																											
LOGICAL*4	LOGICAL																											
INTEGER*2	INTEGER (with /NOI4 specified only)																											
INTEGER*4	INTEGER																											
REAL*4	REAL																											
REAL*8	DOUBLE PRECISION																											
REAL*16																												
COMPLEX*8	COMPLEX																											
COMPLEX*16																												
DOUBLE COMPLEX																												
FLDNAME	F	<p>Structure field is missing a field name.</p> <p>Unnamed fields are not allowed. The effect of an unnamed field can be achieved by using the pseudo-name %FILL in place of a field name in a typed data declaration.</p>																										
FMTEXTCOM	W	<p>Extra comma in format list</p> <p>Example: FORMAT (I4,)</p>																										
FMTEXTNUM	E	<p>Extra number in format list</p> <p>Example: FORMAT (I4,3)</p>																										
FMTINVCHA	E	<p>Format item contains meaningless character</p> <p>An invalid character or a syntax error was detected in a FORMAT statement.</p>																										
FMTINVCON	E	<p>Constant in format item out of range</p> <p>A numeric value in a FORMAT statement exceeds the allowable range. Refer to Chapter 12 for information about range limits.</p>																										
FMTMISNUM	E	<p>Missing number in format list</p> <p>Example: FORMAT (F6.)</p>																										

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
FMTMISSEP	E	Missing separator between format items A required separator character has been omitted between fields in a FORMAT statement.
FMTNEST	E	Format groups nested too deeply Format groups cannot be nested beyond eight levels.
FMTPAREN	E	Unbalanced parentheses in format list The number of right parentheses does not match the number of left parentheses.
FMTSIGN	E	Format item cannot be signed A signed constant is valid only with the P format code.
HOLCURED	E	Count of Hollerith or Radix-50 constant too large, reduced The value specified by the integer preceding the H or R is greater than the number of characters remaining in the source statement.
IDOINVOP	F	Invalid operation in implied-DO list An invalid operation was attempted in an implied-DO list in a DATA statement, for example, a function reference in the subscript or substring expression of an array or character substring reference. Example: <code>DATA (A(SIN(REAL(I))), I=1,10) /101./</code>
IDOINVPAR	F	Invalid DO parameters in implied-DO list An invalid control parameter was detected in an implied-DO list in a DATA statement, for example, an increment of zero.
IDOINVREF	F	Invalid reference to name in implied-DO list A control parameter expression in an implied-DO list in a DATA statement contains a name which is not the name of a control variable of any implied-DO list which has the name in its scope. Example: <code>DATA (A(J), J=1,10),(B(I), I=J,K) /1001./</code> Both J and K in the second implied-DO list are invalid names.

E-12 Diagnostic Messages

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
IDOSYNERR	F	Syntax error in implied-DO list in data initialization Improper syntax was detected in an implied-DO list in data initialization, for instance, improperly nested parentheses.
IMPDECLAR	W	Use of implicit with declaration warnings. An IMPLICIT statement was used in a program compiled with the /WARNINGS=DECLARATIONS qualifier in the FORTRAN command.
IMPMULTYP	E	Letter mentioned twice in IMPLICIT statement, last type used A letter has been given an implicit data type more than once. The last data type given is used.
IMPNONE	E	Untyped name, must be explicitly typed The displayed name has not been defined in any data type declaration statement, and IMPLICIT NONE statement has been specified. Check that the name was not accidentally created by an undetected syntax error. Example: <pre>DO 10 I = 1,10</pre> The apparent DO statement is really an assignment to the accidentally created variable DO10I.
IMPSYNERR	E	Syntax error in IMPLICIT statement Improper syntax was used in an IMPLICIT statement. Refer to Section 8.8 for the syntax rules.
INCDONEST	F	DO or IF statement incorrectly nested One of the following conditions was found: <ul style="list-style-type: none"> A statement label specified in a DO statement has been used previously. Example: <pre>10 I = I + 1 J = J + 1 DO 10 K=1,10</pre>

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
INCDONEST (Cont.)	F	<ul style="list-style-type: none"> A DO loop contains an incomplete DO loop or IF block. Examples: <ol style="list-style-type: none"> <pre> DO 10 I=1,10 J = J + 1 DO 20 h=1,10 J = J + K 10 CONTINUE </pre> <p>The start of the incomplete IF block can be a block IF, ELSE IF, or ELSE statement.</p> <pre> DO 10 I=1,10 J = J + 1 IF (J,GT,20) THEN J = J - 1 ELSE J = J + 1 10 CONTINUE END IF </pre>
INCFILNES	F	<p>INCLUDE files and/or DICTONARY statements nested too deeply</p> <p>Up to 10 levels of nested INCLUDE files and/or DICTONARY statements are permitted.</p>
INCFUNTYP	F	<p>Inconsistent function data types</p> <p>The function name and entry points in a function subprogram must be consistent within one of three groups of data types:</p> <p>Group 1: All numeric types except REAL*16, COMPLEX*16</p> <p>Group 2: REAL*16, COMPLEX*16</p> <p>Group 3: Character</p> <p>Example:</p> <pre> CHARACTER*15 FUNCTION I REAL*4 G ENTRY G </pre>
INCLABUSE	F	<p>Inconsistent usage of statement label</p> <p>Labels of executable statements have been confused with labels of FORMAT statements or with labels of nonexecutable statements.</p> <p>Example:</p> <pre> GO TO 10 10 FORMAT (15) </pre>

E-14 Diagnostic Messages

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
INCLENMOD	F	<p>Incorrect length modifier in declaration</p> <p>An unacceptable length has been specified in a data type declaration (see Section 8.4). For example:</p> <pre>INTEGER PIPES*8</pre>
INCMODNAM	F	<p>Module name not found in library</p> <p>The module name specified in an INCLUDE statement could not be located in the specified library. Check the name of the module and library.</p>
INCOPEFAI	F	<p>Open failure on INCLUDE file</p> <p>The specified file could not be opened, possibly due to an incorrect file specification, nonexistent file, unmounted volume, or a protection violation.</p>
INCSTAFUN	E	<p>Inconsistent statement function reference</p> <p>The actual argument(s) in a statement function reference do not agree in either order, number, or data type with the formal arguments declared.</p>
INCSYNERR	F	<p>Syntax error in INCLUDE file specification</p> <p>The file-name string is not acceptable (invalid syntax, invalid qualifier, undefined device, and so on).</p>
INQUNIT	F	<p>Missing or invalid use of UNIT or FILE specifier in INQUIRE statement.</p> <p>An INQUIRE statement must have a UNIT specifier or a FILE specifier, but may not have both.</p>
INTFUNARG	E	<p>Arguments incompatible with intrinsic function, 'assumed EXTERNAL</p> <p>A function reference was made, using an intrinsic function name, but the argument list does not agree in order, number, or type with the intrinsic function requirements. The function is assumed to be supplied by you as an EXTERNAL function.</p>
INTVALREQ	F	<p>Non-integer expression where integer value required</p> <p>An expression that must be of type INTEGER was another data type.</p>
INVACTARG	E	<p>Invalid use of intrinsic function name as actual argument</p> <p>A generic intrinsic function name was used as an actual argument.</p>

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
INVASSVAR	E	Invalid ASSOCIATEVARIABLE specification An ASSOCIATEVARIABLE specification in an OPEN or DEFINE FILE statement was a dummy argument or an array element.
INVCHAUSE	E	Invalid character used in constant An invalid character was detected in a constant. Valid characters are: Hexadecimal: 0 - 9, A - F, a - f Octal: 0 - 7 Radix-50: A Z, 0 9, \$, period, or space For Radix-60, a space is substituted for the invalid character. For hexadecimal and octal, the entire constant is set to zero.
INVCONSTI	E	Arithmetic error while evaluating constant or constant expression The specified value of a constant is too large or too small to be represented.
INVCONSTR	F	Invalid control structure using ELSE IF, ELSE, or END IF The order of ELSE IF, ELSE, or END IF statements is incorrect. ELSE IF, ELSE, and END IF statements cannot stand alone. ELSE IF and ELSE must be preceded by either a block IF statement or an ELSE IF statement. END IF must be preceded by either a block IF, ELSE IF, or ELSE statement. Examples: 1. DO 10 I=1,10 J = J + I ELSE IF (J,LE,K) THEN Error: ELSE IF preceded by a DO statement. 2. IF (J,LT,K) THEN J = I + J ELSE J = I - J ELSE IF (J,EQ,K) THEN END IF Error: ELSE IF preceded by an ELSE statement
INVDOTERM	W	Statement cannot terminate a DO loop The terminal statement of a DO loop cannot be a GO TO, arithmetic IF, RETURN, block IF, ELSE, ELSE IF, END IF, DO, or END statement.

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
INVENDKEY	W	Invalid END= keyword, ignored The END keyword was used illegally in a WRITE, REWRITE, direct access READ, or keyed access READ statement.
INVENTORY	E	ENTRY within DO loop or IF block, statement ignored An ENTRY statement is not allowed within the range of a DO loop or IF block.
INVEQVCOM	F	Invalid equivalence of two variables in common Variables in common cannot be equivalenced to each other.
INVFUNUSE	F	Invalid use of function name in CALL statement A CALL statement referred to a subprogram name that was used as a CHARACTER, REAL*16, or COMPLEX*16 function. Example: IMPLICIT CHARACTER*10(C) CSCAL =CFUNC(X) CALL CFUNC(X)
INVINIVAR	E	Invalid initialization of variable not in common An attempt was made, in a RLOCK DATA subprogram, to initialize a variable that is not in a common block.
INVINTFUN	E	Name used in INTRINSIC statement is not an intrinsic function A function name which appeared in the INTRINSIC statement is not an intrinsic function.
INVIOSPEC	F	Invalid I/O specification for this type of I/O statement A syntax error was found in the portion of an I/O statement that precedes the I/O list. Examples: 1. TYPE (B),J 2. WRITE(00),J
INVKEYOPE	F	Incorrect keyword in OPEN, CLOSE, or INQUIRE statement An OPEN, CLOSE, or INQUIRE statement contains a keyword which is not valid for that statement.

Table E-1 (Cont.): Source **Program** Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
INVLEFSID	F	Left side of assignment must be variable or array element The symbolic name to which the value of an expression is assigned must be a variable, array element, or character substring reference.
INVLEXEME	F	Variable name, constant, or expression invalid in this context An entity has been used incorrectly; for example, the name of a subprogram was used where an arithmetic expression is required.
INVLOGIF	F	Statement cannot appear in logical IF statement A logical IF statement must not contain a DO statement or another logical IF, IF THEN, ELSE IF, ELSE, END IF, or END statement.
INVNMLELE	F	Invalid NAMELIST element Dummy argument or element other than variable or array name appeared in NAMELIST declaration.
INVNUMSUB	F	Number of subscripts does not match array declaration More or fewer dimensions than were declared for the array are referenced.
INVPERARG	F	Invalid argument to %VAL, %REF, %DESCR, or %LOC The argument specified for one of the built-in functions is not valid. Examples: 1. %VAL (3.5D0) — Argument cannot be REAL*8, REAL*16, character, or complex. 2. %LOC (X+Y) — Argument must not be an expression.
INVTERRUSE	E	%VAL, %REF, or %DESCR used in invalid context The argument list built-in functions (%VAL, %REF, %DESCR) cannot be used outside an actual argument list. Example: X = %REF(Y)
INVQUAL	I	Invalid qualifier or qualifier value in OPTIONS statement An invalid qualifier or qualifier value was specified in the OPTIONS statement. The qualifier is ignored.

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
INVRECUSE	F	<p>Invalid use of record or array name</p> <p>A statement in the program violated one of the following rules:</p> <ul style="list-style-type: none"> • An aggregate cannot be assigned to a nonaggregate or to an aggregate with a structure that isn't the same. • An array name reference cannot be qualified. • Aggregate references cannot be used in I/O lists of formatted I/O statements.
INVREPCOU	E	<p>Invalid repeat count in data initialization, count ignored</p> <p>The repeat count in a data initialization was not an unsigned, nonzero integer constant. The count is ignored.</p>
INVSBSREF	E	<p>Substring reference used in invalid context</p> <p>A substring reference to a variable or array that is not of type CHARACTER has been detected. Example:</p> <pre>REAL X(10) Y = X(J:K)</pre>
INVSTALAB	W	<p>Invalid statement label ignored</p> <p>An improperly formed statement label (namely, a label containing letters) has been detected in columns 1 to 5 of an initial line. The statement label is ignored.</p>
INVSUBREF	F	<p>Subscripted reference to non-array variable</p> <p>A variable that is not defined as an array cannot appear with subscripts.</p>
INVTYPUSE	F	<p>Name previously used with conflicting data type</p> <p>A data type was assigned to a name that had already been used in a context that required a different data type.</p>
IODUPKEY	F	<p>Duplicated keyword in I/O statement</p> <p>Each keyword subparameter in an I/O statement or auxiliary I/O statement can be specified only once.</p>

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
IOINVFMT	F	<p>Format specifier in error</p> <p>The format specifier in an I/O statement is invalid. It must be one of the following:</p> <ul style="list-style-type: none"> • The label of a FORMAT statement. • An asterisk (*). (List-directed I/O.) • A run-time format specifier: variable, array element, or character substring reference. • An integer variable that has been assigned a FORMAT label by an ASSIGN statement.
IOINVKEY	F	<p>Invalid keyword for this type of I/O statement</p> <p>An I/O statement contains a keyword which cannot be used with that type of I/O statement.</p>
IOINVLIST	F	<p>Invalid I/O list element for input statement</p> <p>An input statement I/O list contains an invalid element, such as an expression or a constant.</p>
IOSYNERR	F	<p>Syntax error in I/O list</p> <p>Improper syntax was detected in an I/O list.</p>
LABASSIGN	F	<p>Label in ASSIGN statement exceeds INTEGER*2 range</p> <p>A label whose value is assigned to an INTEGER*2 variable by an ASSIGN statement must not be separated by more than 32K bytes from the beginning of the code for the program unit.</p>
LENCHAFLJN	E	<p>Length specified must match CHARACTER FUNCTION declaration</p> <p>The length specifications for all ENTRY names in a character function subprogram must be the same. Example:</p> <pre>CHARACTER*15 FUNCTION F CHARACTER*20 G ENTRY G</pre>
LOGVALREQ	F	<p>Non-logical expression where logical value required</p> <p>An expression that must be of type LOGICAL was of another data type.</p>

E-20 Diagnostic Messages

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
LOWBOUGRE	E	Lower bound greater than upper bound in array declaration The upper bound of a dimension declarator must be equal to or greater than the lower bound.
MINDIGITS	W	CDD description specifies precision less than allowed for data type. Minimum precision has been supplied. Some Common Data Dictionary data types specify a number of digits which is incompatible with FORTRAN data types. The FORTRAN compiler has expanded the data type to conform to a FORTRAN data type. No action required.
MINOCCURS	I	CDD description contains Minimum Occurs attribute (ignored). FORTRAN does not support the Common Data Dictionary Minimum Occurs attribute. No action required.
MISSAPOS	E	Missing apostrophe in character constant A character constant must be enclosed by apostrophes.
MISSCONST	F	Missing constant A required constant was not found.
MISSDEL	F	Missing operator or delimiter symbol Two terms of an expression are not separated by an operator, or a punctuation mark (such as a comma) has been omitted. Examples: 1. CIRCUM = 3,14 DIAM 2. IF (I 10 , 20 , 30
MISSEND	E	Missing END statement, END is assumed An END statement was missing at the end of the last input file, and it has been inserted.
MISSEXPO	E	Missing exponent after E, D, or Q A floating-point constant was specified in E, D, or Q notation, but the exponent was omitted.
MISSKEY	F	Missing keyword A required keyword, such as TO, was omitted from a statement such as ASSIGN 10 TO I.
MISSLABEL	F	Missing statement label A required statement label reference was omitted.

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
MISSNAME	F	Missing variable or subprogram name A required variable name or subprogram name was not found.
MISSUNIT	F	Unit specifier keyword missing in I/O statement An I/O statement must include a unit specifier subparameter.
MISSVAR	F	Missing variable or constant An expression, or a term of an expression, has been omitted. Examples: 1. WRITE () 2. DIST = *TIME
MULDECNAM	F	Multiple declaration of name A name appears in two or more inconsistent declaration statements.
MULDECTYP	E	Multiple declaration of data type for variable, first type used A variable appears in more than one data type declaration statement. The first type declaration is used.
MULDEFLAB	E	Multiple definition of statement label, second ignored The same label appears on more than one statement. The first occurrence of the label is used.
MULFLDNAM	F	Multiply defined field name Each field name within the same level of a given structure declaration must be unique.
MULSTRNAM	F	Multiply defined STRUCTURE name A STRUCTURE name must be unique among STRUCTURE names.
NAMTOOLON	W	Name longer than 31 characters A symbolic name has been truncated to 31 characters.
NMLIOLIS'I	E	I/O list not permitted with namelist I/O An I/O statement with a namelist specifier incorrectly contained an I/O list.

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
NODFLOAT	W	CDD description specifies the D-Floating data type. The data cannot be represented when compiling /G-FLOAT. D__floating datatype was specified when compiling with /G__FLOATING qualifier. Ignore the warning message or recompile the program using the /NOG__FLOATING qualifier.
NOGFLOAT	W	CDD description specifies G__Floating data type. The data cannot be represented when compiling /NOG__FLOAT. G__floating datatype was specified when compiling with /NOG__FLOATING qualifier. Ignore the warning message or recompile the program using the /G-FLOAT qualifier.
NOHFLOAT	W	CDD description specifies H-Floating data type. The data cannot be represented when compiling /NOG__FLOAT. H__floating datatype was specified when compiling with /NOG__FLOATING qualifier. Ignore the warning message or recompile the program using the /G-FLOATING qualifier.
NONCONSUB	F	Non-constant subscript where constant required Subscript and substring expressions used in DATA and EQUIVALENCE statements must be constants.
NOPATH	W	No path to this statement Program control cannot reach this statement. The statement is deleted. Example: <pre> 10 I = I + 1 GO TO 10 STOP </pre>
OPEDOLOOP	F	Unclosed DO loop or IF block The terminal statement of a DO loop or the END IF statement of an IF block was not found. Example: <pre> DO 20 I=1,10 X = Y END </pre>
OPENOTPER	F	Operation not permissible on these data types An invalid operation was specified, such as an .AND. of two real variables.

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
PROSTOREQ	F	Program storage requirements exceed addressable memory The storage space allocated to the variables and arrays of the program unit exceeds the addressing range of the machine.
REDCONMAR	W	Redundant continuation mark ignored A continuation mark was detected where an initial line is required. The continuation mark is ignored.
REFERENCE	I	CDD description contains Reference attribute (ignored). The Reference attribute is not supported by FORTRAN. No action required.
SOURCETYPE	I	CDD description contains Source Type attribute (ignored). FORTRAN does not support the Common Data Dictionary Source Type attribute. No action required.
STAENDSTR	F	Statement not allowed within structure; structure definition closed A statement not allowed in a structure declaration block was encountered. The compiler assumes that you omitted one or more END STRUCTURE statements.
STAINVSTR	E	Statement not allowed within structure definition; statement ignored A statement not allowed in a structure declaration block was encountered. Structure declaration blocks can only include the following statements: typed data declaration statements, RECORD statements, UNION/END UNION statements, MAP/END MAP statements, and STRUCTURE/END STRUCTURE statements.
STANOTVAL	E	Statement not valid in this program unit, statement ignored A program unit contains a statement that is not allowed; for example, a BLOCK DATA subprogram contains an executable statement.
STAOUTORD	E	Statement out of order, statement ignored The order of statements was not as specified in Section 5.2.2.1. The statement found to be out of order is ignored.
STATOOCOM	F	Statement too complex A statement is too complex to be compiled. It must be subdivided into two or more statements.

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
STRCONTRU	E	String constant truncated to maximum length A character or Hollerith constant can contain up to 2000 characters. A Radix-50 constant can contain up to 12 characters.
STRDEPTH	F	STRUCTUREs/UNIONs/MAPs nested too deeply The combined nesting level limit for structures, unions, and maps is 20 levels.
STRNAME	E	Outer level structure is missing a structure name An outer level STRUCTURE statement must have a structure name in order for a RECORD statement to be able to reference the structure declaration.
STRNOTDEF	F	Structure name in RECORD statement not defined Either a RECORD statement did not contain a structure name enclosed within slashes or the structure name contained in a RECORD statement was not defined in a structure declaration.
SUBEXPVAL	E	Subscript or substring expression value out of bounds An array element beyond the specified dimensions, or a character substring outside the specified bounds, was referenced.
TAGVARIAB	I	CDD description contains Tag Variable attribute (ignored). FORTRAN does not support the Common Data Dictionary Tag Variable attribute. No action required.
TOOMANCOM	F	Too many named common blocks Reduce the number of named common blocks.
TOOMANCON	E	Too many continuation lines, remainder ignored Up to 99 continuation lines are permitted, as determined by the /CONTINUATIONS=n qualifier (default, 19).
TOOMANDIM	E	More than 7 dimensions specified, remainder ignored An array can be defined as having up to seven dimensions.
TOOMANYDO	F	DO and IF statements nested too deeply DO loops and block IF statements cannot be nested beyond 20 levels.

Table E-1 (Cont.): Source Program Diagnostic Messages

Mnemonic	Error Code	Text/Meaning
UNDARR	F	Undimensioned array or statement function definition out of order Either a statement function definition was found among executable statements or <i>an</i> assignment statement involving an undimensioned array was found.
UNDSTALAB	F	Undefined statement label Reference has been made to a statement label that is not defined in the program unit.
UNSUPPTYPE	I	CDD description specifies an unsupported data type. The Common Data Dictionary description for a structure item has attempted to use a data type that is not supported by FORTRAN. The FORTRAN compiler makes the data type accessible by declaring it as an inner structure containing a single CHARACTER %FILL field with an appropriate length. Change the data type to one that is supported by FORTRAN or use the FORTRAN built-in functions to manipulate the contents of the field.
VARINCEQV	F	Variable inconsistently <i>equivalenced</i> to itself EQUIVALENCE statements specify inconsistent relationships between variables or array elements. Example: EQUIVALENCE (A(1), A(2))
ZERLENSTR	E	Zero-length string The length specified for a character, Hollerith, hexadecimal, octal, or Radix-50 constant must not be zero.

E.1.2 Compiler-Fatal Diagnostic Messages

Conditions can be encountered of such severity that compilation must be terminated at once. These conditions are, caused by hardware errors, software errors, and errors that require changing the FORTRAN command. Printed messages have the form:

FORT-F-error name, error message

The first line of the message contains the appropriate file specification **or keyword involved** in the error. The operating system **supplies more** specific information about the error whenever possible. For example, a file read error might produce the following error message:

```
XFORT-F-READERR, error reading _DBA0:[SMITH]MAIN.FOR;3
-RMS-W-RTB, 512 byte record too big for user's buffer
-FORT-F-ABORT, abort
```

Table E-Z lists the diagnostic messages that report the occurrence of such compiler-fatal errors. Because the exact content of the message depends upon the individual problem, only the first line of the message is provided here. Also, “file-spec” represents placement of the actual file specification in the message, and “keyword-value” represents the specific keyword value.

Table E-Z: Compiler-Fatal Diagnostic Messages

I/O Errors

FORT-F-OPENIN, error opening “file-spec” as input
FORT-F-NOSOUIFILE, no source file specified
FORT-F-OPENOUT, error opening “file-spec” as output
FORT-F-READERR, error reading “file-spec”
FORT-F-WRITEERR, error writing “file-spec”
FORT-F-CLOSEIN, error closing “file-spec” as input
FORT-F-CLOSEOUT, error closing “file-spec” as output

Command Qualifier Messages

FORT-F-VALERR, specified value is out of legal range
FORT-F-BADVALUE, “keyword-value” is an invalid keyword value
FORT-F-SUBNOTALL, subqualifier not allowed with negated qualifier

Compiler Internal Logic Error

FORT-F-BUGCHECK, internal consistency failure

If you receive the compiler internal logic error, FORT-F-BUGCHECK, you should report both the error and the circumstance in which it occurred to DIGITAL by means of a Software Performance Report (SPR).

E.1.3 Compiler limits

There are limits to the size and complexity of a single VAX FORTRAN program unit. There are also limits on the complexity of FORTRAN statements. Table E-3 describes some of these limits.

Table E-3: Compiler Limits

Language Element	Limit
Structure nesting	20
DO and block IF statement nesting (combined)	20
Actual number of arguments per CALL or function reference	255
Named common blocks	250
Format group nesting	8
Labels in computed or assigned GO TO list	500
Parentheses nesting in expressions	40
INCLUDE file nesting	10
Continuation lines	99
FORTTRAN source line length	132 characters
Symbolic name length	31 characters
constants	
Character, Hollerith	2000 characters
Radix-50	12 characters
Array dimensions	1
Number of names in a NAMELIST group	250

The amount of data storage, the size of arrays, and the total size of executable programs are limited only by the amount of process virtual address space available, as determined at VAX/VMS system generation.

E.2 Run-Time Diagnostic Messages

Errors that occur during execution of your FORTRAN program are reported by diagnostic messages from the Run-Time Library. These messages may result from hardware conditions, file system errors, errors detected by RMS, errors that occur during transfer of data between the program and an internal record, computations that cause overflow or underflow, incorrect calls to the Run-Time Library, problems *in* array descriptions, and conditions detected by the operating system. Refer to the *VAX/VMS Run-Time Library Routines Reference Manual* for more information.

E.2.1 Run-Time Library Diagnostic Message Presentation

Run-Time Library diagnostic messages are usually sent either to your terminal (interactive mode) or to the log file (batch mode).

E-28 Diagnostic Messages

E.2.2 Run-Time library Diagnostic Messages

Descriptions of Run-Time Library diagnostic messages can be found in Table E-4.

Table E-4 lists each Run-Time diagnostic message in alphabetical order according to its unique 6- to g-character name. For organizational purposes, the message prefixes **FOR\$**, **SS\$**, and **MTH\$** are not shown in this table. (Note: Refer to Table 18-1 for a presentation of the messages in order of their error numbers.)

Table E-4: Run-Time Diagnostic Messages

Condition Symbol	Err NO	Sev	Message Text
ADJARRDIM	93	F,C	adjustable array dimension error Upon entry to a subprogram, the evaluation of dimensioning information detected an array in which: <ul style="list-style-type: none"> • An upper-dimension bound was less than a lower-dimension bound. • The dimensions imply an array that exceeds the addressable memory.
ATTACCNON	36	F	attempt to access non-existent record One of the following conditions occurred: <ul style="list-style-type: none"> • An attempt was made to READ, FIND, or DELETE a nonexistent record from a relative organization file using direct access. • An attempt was made to access beyond the end of the file with a direct access READ or FIND to a sequential organization file. • An attempt was made to read a nonexistent record from an indexed organization file with a keyed access READ statement.
BACERR	23	F	BACKSPACE error One of the following conditions occurred: <ul style="list-style-type: none"> • The file was not a sequential organization file. • The file was not opened for sequential access. (A unit opened for append access may not be backspaced until a REWIND statement is executed for that unit.) • RMS detected an error condition during execution of a BACKSPACE statement.

Table E-4 (Cont.): Run-Time Diagnostic Messages

Condition Symbol	Err NO	SW	Message Text
CLOERR	28	F	CLOSE error An error condition was detected by RMS during execution of a CLOSE statement.
DELERR	55	F	DELETE error One of the following conditions occurred: <ul style="list-style-type: none"> • On a direct access DELETE, the file did not have relative organization. • On a current record DELETE, the file did not have relative or indexed organization, or the file was opened for direct access. • RMS detected an error condition during execution of a DELETE statement.
DUPFILSPE	21	F	duplicate file specifications Multiple attempts were made to specify file attributes without an intervening close operation. One of the following conditions occurred: <ul style="list-style-type: none"> • A DEFINE FILE was followed by DEFINE FILE. • A DEFINE FILE was followed by an OPEN statement. • A CALL ASSIGN or CALL FDBSET was followed by an OPEN statement.
ENDDURREA	24	F	end-of-file during read One of the following conditions occurred: <ul style="list-style-type: none"> • An RMS end-of-file condition was encountered during execution of a READ statement that did not contain an END, ERR, or IOSTAT specification. • An end-of-file record written by the ENDFILE statement was encountered during execution of a READ statement that did not contain an END, ERR, or IOSTAT specification. • An attempt was made to read past the end of an internal file character string or array during execution of a READ statement that did not contain an END, ERR, or IOSTAT specification.

E-30 Diagnostic Messages

Table E-4 (Cont.): Run-Time Diagnostic Messages

Condition Symbol	Err No	Sev	Message Text
FLTDIV	73	F,C	<p>arithmetic trap, zero divide</p> <p>During a floating-point or decimal arithmetic operation, an attempt was made to divide by 0.0. If floating, the result of the operation was set to the reserved operand, -0. If decimal, the result of the operation is unpredictable.</p>
FLTDIV-F	73	F,C	<p>arithmetic fault, zero divide</p> <p>During a floating-point arithmetic operation, an attempt was made to divide by zero.</p>
FLTOVF	72	F,C	<p>arithmetic trap, floating overflow</p> <p>During an arithmetic operation a floating-point value exceeded the largest representable value for that data type. The result of the operation was set to the reserved operand, -0.</p>
FLTOVF-F	72	F,C	<p>arithmetic fault, floating overflow</p> <p>During an arithmetic operation, a floating-point value exceeded the largest representable value for that data type.</p>
FLTUND	74	F,C	<p>arithmetic trap, floating underflow</p> <p>During an arithmetic operation a floating-point value became less than the smallest representable value for that data type and was replaced with a value of zero.</p>
FLTUND-F	74	F,C	<p>arithmetic fault, floating overflow</p> <p>During an arithmetic operation a floating-point value became less than the smallest representable value for that data type.</p>
FORVARMIS	61	F,C	<p>format/variable-type mismatch</p> <p>An attempt was made either to read or write a real variable with an integer field descriptor (I or L), or to read or write an integer or logical variable with a real field descriptor (D, E, F, or G). If execution continued, the following actions occurred:</p> <ul style="list-style-type: none"> • If I or L, convert as if INTEGER*4. • If D, E, F, or G, convert as if REAL*4.

Table E-4 (Cont.): Run-Time Diagnostic Messages

Condition Symbol	Err NO	Sev	Message Text
ENDFILERR	33	F	<p>ENDFILE error</p> <p>One of the following conditions occurred:</p> <ul style="list-style-type: none"> . The file was not a sequential organization file with variable-length records. • The file was not opened for sequential or append access. . An unformatted file did not contain segmented records. • RMS detected an error during execution of an ENDFILE statement.
ERRDURREA	39	F	<p>error during read</p> <p>RMS detected an error condition during execution of a READ statement.</p>
ERRDURWRI	38	F	<p>error during write</p> <p>RMS detected an error condition during execution of a WRITE statement.</p>
FILNAMSP	43	F	<p>file name specification error</p> <p>A file-name specification given to OPEN, INQUIRE, or CALL ASSIGN was not acceptable to RMS.</p>
FILNOTFOU	29	F	<p>file not found</p> <p>A file with the specified name could not be found during an open operation.</p>
FINERR	67	F	<p>FIND error</p> <p>RMS detected an error condition during execution of a FIND statement.</p>
FLOOVEMAT	88	F,C	<p>floating overflow in math library</p> <p>A floating overflow condition was detected during execution of a math library procedure. The result returned was the reserved operand, -0.</p>
FLOUNDMAT	89	F,C	<p>floating underflow in math library</p> <p>A floating underflow condition was detected during execution of a math library procedure. The result returned was zero.</p>

Table E-4 (Cont.): Run-Time Diagnostic Messages

Condition Symbol	Err No	Sev	Message Text
INCFILORG	51	F	<p>inconsistent file organization</p> <p>One of the following conditions occurred:</p> <ul style="list-style-type: none"> • The file organization specified in an OPEN statement did not match the organization of the existing file. • The file organization of the existing file was inconsistent with the specified access mode; that is, direct access was specified with an indexed organization file, or keyed access was specified with a sequential-or relative organization file.
INCKEYCHG	50	F	<p>inconsistent key change or duplicate key</p> <p>A WRITE or REWRITE to an indexed organization file caused a key field to change or be duplicated. This condition was not allowed by the attributes of the file, as established when the file was created.</p>
INCOPECLO	46	F	<p>inconsistent OPEN/CLOSE parameters</p> <p>Specifications in an OPEN or CLOSE statement were inconsistent. Some invalid combinations are:</p> <ul style="list-style-type: none"> • READONLY with STATUS= 'NEW' or STATUS='SCRATCH' • ACCESS='APPEND' with READONLY, STATUS='NEW', or STATUS='SCRATCH' • DISPOSE='SAVE', 'PRINT', or 'SUBMIT' with STATUS='SCRATCH' • DISPOSE='DELETE' with READONLY
INCRECLEN	37	F	<p>inconsistent record length</p> <p>One of the following conditions occurred:</p> <ul style="list-style-type: none"> • An attempt was made to create a new relative, indexed, or direct access file without specifying a record length. • An existing file was opened in which the record length did not match the record size given in an OPEN or DEFINE FILE statement.

Table E-4 (Cont.): Run-Time Diagnostic Messages

Condition Symbol	Err No	Sev	Message Text
INCRECTYP	44	F	inconsistent record type The RECORDTYPE value in an OPEN statement did not match the record type attribute of the existing file that was opened.
INFFORLOO	60	F	infinite format loop The format associated with an I/O statement that included an I/O list had no field descriptors to use in transferring those values.
INPCONERR	64	F,C	input conversion error During a formatted input operation, an invalid character was detected in an input field, or the input value overflowed the range representable in the input variable. The value of the variable was set to zero.
INPRECTOO	22	F	input record too long A record was read that exceeded the explicit or the default record length specified at OPEN (or by the default OPEN). To read the file, use an OPEN statement with a RECL value of the appropriate size.
INPSTAREQ	67	F	input statement requires too much data An unformatted READ statement attempted to read more data than existed in the record being read.
INSVIRMEM	41	F	insufficient virtual memory The FORTRAN Run-Time Library attempted to exceed its virtual page limit while dynamically allocating space.
INTDIV	71	F,C	arithmetic trap, integer zero divide During an integer arithmetic operation, an attempt was made to divide by zero. The result of the operation was set to the dividend, which is equivalent to division by one.
INTOVF	70	F,C	arithmetic trap, integer overflow During an arithmetic operation, an integer value exceeded byte, word, or longword range. The result of the operation was the correct low-order part.

Table E-4 (Cod.): Run-Time Diagnostic Messages

Condition Symbol	Err NO	Sev	Message Text
INVARGFOR	48	F	<p>invalid argument to FORTRAN Run-Time Library</p> <p>One of the following conditions occurred:</p> <ul style="list-style-type: none"> • An invalid argument was given to a PDP-11 FORTRAN compatibility subroutine such as ERRSET. • The FORTRAN compiler passed an invalid coded argument to the Run-Time Library. This can occur if the compiler is newer than the Run-Time Library in use.
INVARGMA'I	81	F	<p>invalid argument to math library</p> <p>One of the mathematical procedures detected an invalid argument value.</p>
INVKEYSPE	49	F	<p>invalid key specification</p> <p>A key specification in an OPEN statement or in a keyed access READ statement was invalid. For example, the key length may have been zero or greater than 255 bytes, or the key length may not conform to the key specification of the existing file.</p>
INVLOGUNI	32	F	<p>invalid logical unit number</p> <p>A logical unit number greater than 99 or less than zero was used in an I/O statement.</p>
INVREFVAH	19	I'	<p>invalid reference to variable in NAMELIST input</p> <p>The variable in error is shown as "varname" in the message text. One of the following conditions occurred:</p> <ul style="list-style-type: none"> • The variable was not a member of the namelist group. • An attempt was made to subscript the scalar variable. • A subscript of the array variable was out-of-bounds. • There were too many or too few subscripts for the variable.



24272231 7

