

Exercise answers

The answers are in exercise-number / chapter-number order. This unusual ordering will help you to avoid seeing the answer to the following exercise.

Exercises 1

1.1 A card reader is used to input programs and data.

2.1 I NUMBER LETTER

3.1 (1) 3 (FORMAT statements are not executed)

(2) DO 30 I=1,85

4.1 SUMDSQ=0.0

DO 25 I=1,5

SUMDSQ=SUMDSQ+(SMARK(I)-AVRAGE)**2

25 CONTINUE

SD=(SUMDSQ/5.0)**0.5

5.1 (1) 0.8415

(2) $1 + 2.1170 = 3.1170$

(3) $\log_e(1 - 1.6487)$ - no result as ALOG argument must be positive.

(4) $0.4621 + 0.7854 = 1.2475$

6.1 190

7.1 VOLUME=RELENTH*AREA(DIAM/2.0)

8.1 Change the IF statement to IF (ITOWN.NE.INTOWN) GOTO 30

9.1 IF (ICOUNT.EQ.1) GOTO 5

IF (ICOUNT.EQ.2) GOTO 20

IF (ICOUNT.EQ.3) GOTO 30

IF (ICOUNT.EQ.4) GOTO 25

IF (ICOUNT.EQ.5) GOTO 40

10.1 EQUIVALENCE (M,N), (A,B,C,AR(6)), (AR(7),ROOT(23))

Exercises 2

1.2 So that it can be converted to instructions which a computer is able to execute - a computer is not able to directly execute a FORTRAN program.

2.2 1 and 3

```
3.2      DO 20 N=1,100
          NSQ=N*N
          NCUBE=NSQ*N
          WRITE(6,10) N,NSQ,NCUBE
10       FORMAT(1H , 3I8)
20       CONTINUE
        STOP
        END
```

4.2 3 5 -2.0

5.2 (1) $\text{SIN}(X+Y)**2+\text{SIN}(X/Y)/\text{COS}(X/Y)$
(2) $\text{ALOG10}(1.0+\text{EXP}(\text{SQRT}(A+B)))$

6.2 (1) 132

```
(2)      ITOT=0
          DO 90 ITEM=1,5
            ITOT=ITOT+NSTOCK(ITEM,NSHOP)
          90 CONTINUE
```

7.2 $\text{VOLUME}(\text{RLENGTH},\text{DIAM})=\text{RLENGTH}*\text{AREA}(\text{DIAM}/2.0)$

8.2 DATA A,R(1),R(2),R(3),I,M,N/4*0.0,3*1/

```
9.2      IVAT  PRICE
-1       undefined but likely to be 100.0
0        100.0
1        108.0
2        110.0
3        125.0
4        undefined but likely to be 100.0
5        undefined but likely to be 100.0
```

10.2 ROOT(22)

Exercises 3

```
1.3      READ(5,10) A,B,C
10       FORMAT(F7.0,F7.0,F5.0)
```

2.3 (1) 3
(2) 1
(3) 0
(4) 0

```

3.3      DO 20 N=1,101,5
          RN=N-1
          RDIV10=RN/10.0
          WRITE(6,10) RDIV10
10       FORMAT(1H ,F10.1)
20       CONTINUE
          STOP
          END

4.3      DIMENSION NOS(20),NOSREV(20)
          READ(5,10) NOS
10       FORMAT(20I4)
          DO 20 I=1,20
            J=21-I
            NOSREV(I)=NOS(J)
20       CONTINUE
          WRITE(6,30) NOSREV
30       FORMAT(1H ,20I5)
          STOP
          END

5.3      IF (X.LT.0.0) X=-X

6.3      ITOT=0
          DO 120 ITEM=1,5
            DO 110 ISHOP=1,4
              ITOT=ITOT+NSTOCK(ITEM,ISHOP)
110      CONTINUE
120     CONTINUE

7.3      VOLPIP=VOLUME(2.0*ROUT,RLENTN)-VOLUME(2.0*RIN,RLENTN)

8.3      DATA M,N,HIGH/1,5,50.0/
          -the variable S needs to be re-initialised during program
          execution.

9.3      IF (A.GE.55.0) GOTO 40
          IF (B.GT.96.0) GOTO 30
40      :

```

Exercises 4

```

1.4      WRITE(6,20) A,B,C
20       FORMAT(1H ,F7.2,F7.2,F5.1)

2.4      IREM is assigned the remainder of the division of M by N.

3.4      (1) So that first time round the loop the first number
          to be summed is added to zero and not to any other
          value.

```

(2) Change ISUM=0 to IPROD=1
 ISUM=ISUM+NUMBER to IPROD=IPROD*NUMBER
 and WRITE(6,30) ISUM to WRITE(6,30) IPROD

(3) READ(5,10) ICENTF,ICENTL,ICENTI
 10 FORMAT(3I5)
 DO 30 ICENT=ICENTF,ICENTL,ICENTI
 CENT=ICENT
 FAH=9.0*CENT/5.0+32.0
 WRITE(6,20) ICENT,FAH
 20 FORMAT(1H ,I5,F6.1)
 30 CONTINUE
 STOP
 END

4.4 So that SMAX has a definite value for the comparison during the first execution of the loop statements.

5.4 $I=M+IFIX(A*(X-Y))$

6.4 DIMENSION NSTOCK(5,4),LEVEL(5)
 :
 DO 190 ITEM=1,5
 DO 180 ISHOP=1,4
 IF (NSTOCK(ITEM,ISHOP).GT.LEVEL(ITEM)) GOTO 180
 WRITE(6,170) ITEM,ISHOP
 170 FORMAT(1H ,2I3)
 180 CONTINUE
 190 CONTINUE

7.4 $EQUAT(X,I,Y,J)=X**I+Y**J$
 and $IVALUE(M,A,B)=M+IFIX(A-B)$

8.4 INTEGER R,S,T,A(2,3)
 REAL M1,M2,M3,K(12),LEN
 DIMENSION M(6),X(4)
 DATA R,M1,M2,M3/5,3*0.0/
 DATA A(1,1),A(1,2),A(1,3)/3*1/
 DATA X(1),X(2),X(3),X(4)/4*0.0/

The arrays M and A could have been specified in the INTEGER and REAL statements, respectively.

There may be more than one of each of the specification statements. The initialisation could have been specified in one DATA statement with a continuation line.

9.4 IF (IDAY.EQ.28.AND.
 1 (MONTH.EQ.1
 2 .OR.MONTH.GE.3.AND.MONTH.LE.12
 3 .OR.MONTH.EQ.2.AND.
 4 IYEAR.EQ.IYEAR/4*4)) IDAY=29

Exercises 5

1.5 76.0 55760.72 1.0 67.5

```
2.5      READ(5,10) HEIGHT,WIDTH,LENGTH
10      FORMAT(2F6.0,I5)
         RLENTH=LENGTH
         VOLUME=HEIGHT*WIDTH*(RLENTH/100.0)
         WRITE(6,20) HEIGHT,WIDTH,LENGTH,VOLUME
20      FORMAT(1H ,2F6.2,I5,F8.2)
         STOP
         END
```

```
3.5      (1)    I=1
              :
              I=I+1
              IF (I.LE.N) GOTO 65
```

(2) once - DO loops are normally executed at least once, regardless of the value of the DO statement parameters, because the test for the end of looping is usually made after the loop statements.

4.5 Change .LE. to .GE. and SMAX to SMIN.

$$5.5 1 + \frac{1}{2} + \frac{1}{8} + \frac{1}{48} = 1\frac{31}{48}$$

```
6.5      WRITE(6,90) (I,(NSTOCK(I,J),J=1,4),I=1,5)
```

7.5 READ statement-the first statement of the main program is always executed first.

```
8.5      CHARACTER*6 N1,N2,TEXT(20)*1
```

```
9.5      .TRUE.
```

Exercises 6

```
1.6      PAGE1    ADRESS
```

```
2.6      IF (ICODE.LE.12) SIZE=SIZE*0.92
```

```
3.6            I=M
135            :
              I=I+L
              IF (I.LE.N) GOTO 135
```

```
4.6            DO 50 I=1,5
              IF (SMARK(I).NE.FINDMK) GOTO 50
              FINDIN=I
              GOTO 80
50      CONTINUE
60      WRITE(6,70)
70      FORMAT(1H ,14HMARK NOT FOUND)
80      :
```

5.6 The I loop counter is required as the denominator in the term calculation and also the DO loop acts as a maximum safeguard on the number of terms that are computed.

```
6.6 WRITE(6,100) (NSTOCK(ITEM,NSHOP),PRICE(ITEM),VALUE(ITEM),ITEM=1,5)
100 FORMAT(1H ,I3,F8.2,F12.2)
```

```
7.6 0.1 3.21 (0.01 + 0.2 + 3.0)
     0.2 3.44 (0.04 + 0.4 + 3.0)
```

8.6 Replace the DIMENSION statement by
CHARACTER*1 LETTER,ISENT(80)
and replace all occurrences of A1 by A.

```
9.6 bbbbbbTbbFbbbbbbT
```

Exercises 7

```
1.7 a -  $\left(\frac{b}{c} \times d\right)$ 
```

```
2.7 IF (TIME.LT.24.00) GOTO 50
     TIME=0.0
     IDAY=IDAY+1
50 :
```

3.7 A maximum of 999 numbers can be dealt with correctly - if there are more, the first 1000 numbers only will be input and the average computed will be incorrect.

```
4.7 WRITE(6,10)
10 FORMAT(1H1,18HTHIS IS A NEW PAGE/
11H0,22HTHIS IS THE THIRD LINE/
21H ,23HTHIS IS THE FOURTH LINE)
```

In order to see more easily what the layout of the output will be it is best to start each line of printer output on a new continuation line in the FORMAT statement.

```
5.7 (1) 1
     (2) 3
     (3) 15
     (4) 3
```

```
6.7 NUMBER 8
     NUMBER 12
     NUMBER 9
     NUMBER 5
     34
```

```

7.7 (1) FUNCTION AREAT(A,B,C)
      S=(A+B+C)/2.0
      AREAT=SQRT(S*(S-A)*(S-B)*(S-C))
      RETURN
      END

      (2) AREAOC=D*D-4.0*AREAT(S,(D-S)/2.0,(D-S)/2.0)
      or AREAOC=16.0*AREAT(S/2.0,D/2.0,SQRT(S*S+D*D)/2.0)

```

```

8.7 CHARACTER*3 ANSWER
      NOES=0
      IYESES=0
      DO 30 I=1,1000
      READ(5,10) ANSWER
10  FORMAT(A)
      IF (ANSWER.EQ.'ZZZ') GOTO 40
      IF (ANSWER.GE.'Y ') GOTO 20
      NOES=NOES+1
      GOTO 30
20  IYESES=IYESES+1
30  CONTINUE
40  WRITE(6,50) NOES,IYESES
50  FORMAT(1H ,2HNO ,3X,I3/1H ,3HYES,2X,I3)
      STOP
      END

```

To cater for the alternatives for each answer the .GE. operator has been used. This will cause the condition ANSWER.LE.'Y ' to be true if ANSWER is either YES or Ybb. Alternatively, the program could have been written to input and check only the first letter.

```

9.7 IF (X-Y) 30,40,20
      or IF (Y-X) 20,40,30

```

Exercises 8

```

1.8 (1) ((X+Y)*(X+Z))**3
      (2) (-B+(B**2-4.0*A*C)**0.5)/(2.0*A)
      (3) ((1.0+R)**(-P))/(P*(P-1.0))

```

2.8 Replace the statement GOTO 190 with

```

      IF (X*X.EQ.Y+Z) GOTO 160
      A=A*0.25
      B=B*0.25
      GOTO 170
160  A=A*0.5
      B=B*0.5
170 GOTO 190

```

```

3.8   WRITE(6,10)
      10 FORMAT(1H1,18HTHIS IS A NEW PAGE)
        WRITE(6,20)
      20 FORMAT(1H0,22HTHIS IS THE THIRD LINE)
        WRITE(6,30)
      30 FORMAT(1H ,23HTHIS IS THE FOURTH LINE)

```

```

4.8   8 lines as follows
      1 blank line
      THIS IS THE MAIN HEADING
      3 blank lines
      THIS IS A SUB-HEADING
      2 blank lines

```

```

5.8   (1) No - NUM should be a constant
      (2) Yes
      (3) No - I*2 must be written as 2*I.
      (4) No - N should be a constant.

```

```

6.8   150

```

```

7.8   FUNCTION NUMGT(ARR,N,VALUE)
      DIMENSION ARR(100)
      NUMGT=0
      DO 10 I=1,N
        IF (ARR(I).GT.VALUE) NUMGT=NUMGT+1
10    CONTINUE
      RETURN
      END

```

```

8.8   IF (N.EQ.1) MAYBE=CAN//'NOT'

```

```

9.8   IF (Y) 10,20,30
      10 X=-99.9
        GOTO 40
      20 X=0.0
        GOTO 40
      30 X=49.9
      40 :

```

Exercises 9

```

1.9   READ(5,10) DIAM1,DIAM2
      10 FORMAT(2F8.0)
      AREA1=3.1416*(DIAM1/2.0)**2
      AREA2=3.1416*(DIAM2/2.0)**2
      AREA=AREA2-AREA1
      WRITE(6,20) AREA1,AREA2,AREA
      20 FORMAT(1H ,3F14.2)
      STOP
      END

```

```

2.9   ISIGN=-1
      IF (X.EQ.0.0) ISIGN=0
      IF (X.GT.0.0) ISIGN=1

```



```

3.9   A- 6 to 13 B- 17 to 21 N- 26 to 31
4.9   30 FORMAT(1H ,F5.1)
5.9   5   4   4   4   4   2   2   2   2   2   1
      )           )           )           )
      4   5   2   2   2   4   3   3   3   1   2
      )           )           )           )
      2   2   5   3   3   3   4   1   1   3   3
      )           )           )           )
      3   3   3   5   1   1   1   4   4   4   4
      )           )           )           )
      1   1   1   1   5   5   5   5   5   5   5

```

```

7.9   FUNCTION ISIGN(A)
      ISIGN=0
      IF (A.GT.0.0) ISIGN=1
      IF (A.LT.0.0) ISIGN=-1
      RETURN
      END

```

```

8.9   NUM=0
      DO 10 I=1,50
      IF(PEOPLE(I)(13:14).LE.'21') NUM=NUM+1
10    CONTINUE

```

```

9.9   IY=Y+2.0
      IF (IY.LT.1.OR.IY.GT.3) GOTO 99
      GOTO(10,20,30), IY
10    X=-99.9
      GOTO 40
20    X=0.0
      GOTO 40
30    X=49.9
40   :

```

Exercises 10

```

2.10  20 FORMAT(1H ,18HTHE ARTICLE CODED ,I3,
      1          15H HAS A SIZE OF ,F8.1)

```

```

3.10  DO 30 PLACE=0,LENTH,LENTH/(NPOSTS-1)
      WRITE(6,20) PLACE
20    FORMAT(1H ,F6.1)
30    CONTINUE

```

```

4.10  WRITE(6,90) (I,NUMIN(I),I=2,20,2)
80    FORMAT(1H ,I2,I3)

```

The original statements output the values on one line whereas the above statements output the two values for each house on separate lines (using the rescanning facility of FORMAT statements).

5.10 If we used the two statements

```
NUMS(K)=NUMS(K+1)
NUMS(K+1) = NUMS(K)
```

to exchange the two values, the first statement would destroy the original value of NUMS(K) so it must first be assigned to a temporary variable.

```
7.10      :
          CALL CIRCLE(D/2.0,AREA,CIRCUM)
          VOLUME=RL*AREA
          SAREA=2.0*AREA+RL*CIRCUM
          :
```

9.10 GOTO (20,30,40,50), IVAT+1

Exercises 11

2.11 AVRAGE=(MARK1+MARK2+MARK3)/3.0

If the sum of the marks had been required elsewhere in the program the statements would not be combined because using a single statement the sum of the marks is not retained in a variable.

3.11 THIS TEXT IS NOW IN ORDER

4.11 10 FORMAT(4I2)

5.11 Replace .LT. by .GT.

```
7.11      SUBROUTINE CHECK(ARR,M,N,NEG,IZERO,IPOS)
          DIMENSION ARR(100,1000)
          NEG=0
          IZERO=0
          IPOS=0
          DO 20 I=1,M
            DO 10 J=1,N
              IF (ARR(I,J).LT.0.0) NEG=NEG+1
              IF (ARR(I,J).EQ.0.0) IZERO=IZERO+1
              IF (ARR(I,J).GT.0.0) IPOS=IPOS+1
            10 CONTINUE
          20 CONTINUE
          RETURN
          END
```

```

9.11    IF (I.EQ.1) THEN
        X=0.0
        Y=0.0
    ELSE IF (I.EQ.3) THEN
        X=5.0
        Y=5.0
    ELSE IF (M+N.LT.99) THEN
        P=0.0
    ELSE
        P=0.0
        Q=0.0
    END IF

```

Exercises 12

2.12 2.0
A**B**C is equivalent to A**(B**C).

```

4.12    DIMENSION MARKS(0:9)
        DO 10 I=0,9
            MARKS(I)=0
10    CONTINUE
        READ(5,20) N
20    FORMAT(I3)
        DO 40 I=1,N
            READ(5,30) MARK
30    FORMAT(I2)
            MARKS(MARK)=MARKS(MARK)+1
40    CONTINUE
        DO 60 I=0,9
            WRITE(6,50) I,MARKS(I)
50    FORMAT(1H ,I2,4X,I3)
60    CONTINUE
        STOP
        END

```

```

5.12    IF (ABS(TERM).LT.0.5E-6) GOTO 20
        WRITE(6,5) TERM
5    FORMAT(1H ,E11.4)

```

```

7.12    (1)        Z(X,Y)=2.0*X*X+Y*Y+3.0
           :
           VAL=Z(A,B)+Z(V,W)

```

```

(2)        FUNCTION Z(X,Y)
           Z=2.0*X*X+Y*Y+3.0
           RETURN
           END
           :
           VAL=Z(A,B)+Z(V,W)

```

```

(3)      SUBROUTINE EQUAT(X,Y,Z)
          Z=2.0*X*X+Y*Y+3.0
          RETURN
          END
          :
          CALL EQUAT(A,B,R1)
          CALL EQUAT(V,W,R2)
          VAL=R1+R2

```

Of course, neither a function nor a subroutine would normally be used for a single expression computation.

9.12 IF (P1SEX.NEQV.P2SEX) DIFF=.TRUE.

Exercises 13

```

2.13     IF (X.GT.100.0) THEN
          XROOT=SQRT(X)
          WRITE(6,250) X,XROOT
250      FORMAT(1H ,2F10.2)
          END IF
          :

```

Notice that the condition has had to be reversed.

- 7.13 (1) the program is more manageable
 (2) each logical unit of the program can be tested more easily
 (3) the subroutines may be usable in other programs.

Exercises 14 and 15

```

2.14     IF (A*A.LE.B*B) THEN
          X=3.0*X
          Y=3.0*Y
          Z=3.0*Z
        ELSE
          X=2.0*X
          Y=2.0*Y
        END IF
          :

```

Notice that the condition has had to be reversed. (Alternatively, the groups of statements could have been interchanged).

```

2.15 20 FORMAT(1H , 'THE ARTICLE CODED ',I3,
1      'HAS A SIZE OF ',F8.1)

```

Notice the use of 1 in column 6 to indicate a continuation of the FORMAT statement.

Index

- .AND. operator 9.2
- Arguments 5.1, 7.2
 - actual 7.2, 7.4
 - dummy 7.2, 7.4
 - input 7.4
 - output 7.4
- Arithmetic IF statement 9.5
- Arrays
 - 1-D 4.1, 4.4
 - 2-D 6.1, 6.4
 - name 4.1
 - subscripts 4.1
 - subscript expressions 5.5, 5.8
- Assignment statement 1.3

- Branching 2.4

- Card reader 1.1
- Central processing unit 1.1
- Character
 - constants 8.5
 - functions 8.5
 - handling 8.1
 - operators 8.5
 - substrings 8.5
- CHARACTER statement 8.5
- Coding form 1.2

- Comments 2.6, 2.7
- COMMON statement 10.2
- Compilation 1.2
- Computed GOTO statement 9.1, 9.6
- Computer systems 1.1
- Continuation statements 2.6
- CONTINUE statement 3.1
- Control statements 2.6

- DATA statement 8.2, 8.5
- Decimal exponent constants 5.7
- DIMENSION statement 4.1
- DO statement 3.1, 3.2, 5.4
- DOUBLE PRECISION statement 8.4

- END statement 1.3
- EQUIVALENCE statement 10.1
- .EQV. operator 9.6
- Exponentiation operator 1.5

- FORMAT statement 1.3, 4.3
 - field descriptors
 - A8.1
 - apostrophe 2.7
 - E 5.7
 - F 1.3
 - H 2.5, 3.2

I 2.3
 L 9.4
 T 3.3
 TL 3.3
 TR 3.3
 X 3.2
 rescanning 4.2
 / 4.2

Functions
 basic external 5.1, 5.8
 intrinsic 5.2, 5.8
 library 5.1, 5.2, 5.8
 standard 5.1, 5.2, 5.8
 ABS (absolute) 5.2, 5.8
 ACOS (arccosine) 5.8
 ALOG (natural logarithm) 5.1
 ALOG10 (common logarithm) 5.1
 ASIN (arcsine) 5.8
 ATAN (arctangent) 5.1
 COS (cosine) 5.1
 COSH (hyperbolic cosine) 5.8
 FLOAT (integer to real) 5.2
 IABS (integer absolute) 5.2
 IFIX (real to integer) 5.2
 INT (real to integer) 5.8
 REAL (integer to real) 5.8
 SIN (sine) 5.1
 SINH (hyperbolic sine) 5.8
 SQRT (square root) 5.1
 TAN (tangent) 5.8
 TANH (hyperbolic tangent) 5.1

FUNCTION statement 7.3
 Function subprograms 7.3
 GOTO statement 2.4
 Group format specification 6.3

Group repeat count 6.3
 IF statement 2.4
 IF-ELSE IF-END IF statement 9.6
 IF-THEN-ELSE-END IF statement 2.7
 IF-THEN-END IF statement 2.7
 Implied-DO list 4.3, 6.2, 4.4
 IMPLICIT statement 8.5
 Initial values 8.2
 Input device 1.1
 Integer
 constants 2.1
 variables 2.1
 variable names 2.1
 INTEGER statement 8.3
 Line controls 3.2
 Line printer 1.1
 Lists 4.1
 Logical
 constants 9.3
 expressions 9.2
 operators 9.2
 statements 9.3
 variables 9.3
 Looping 3.1
 Loop statements 3.1
 Loop variable 3.1, 3.3
 Machine code 1.2
 Main program 7.1
 Modular programming 7.4
 .NEQV. operator 9.6
 Nested DO loops 5.4
 .NOT. operator 9.2

Operators

- arithmetic 1.5
- logical 9.2
- relational 2.4
- .OR. operator 9.2
- Output device 1.1
- Output layout 3.2

- PARAMETER statement 5.8
- Precision 8.4
- Procedures 7.1
- Programming 1.2

- READ statement 1.3, 4.3, 3.3
- Real
 - constants 1.4
 - variables 1.4
 - variable names 1.4
- REAL statement 8.3

Relational

- expressions 2.4
- operators 2.4
- RETURN statement 7.3

- Sorting 5.6
- Specification statements 8.3
- Statements 2.6
- Statement functions 7.2
- Statement numbers 2.6
- Stop numbers 3.1
- STOP statement 1.3
- Successive approximation 5.3
- SUBROUTINE statement 7.4
- Subroutine subprograms 7.4

- Text output 2.5
- Type-statements 8.3

- WRITE statement 1.3, 4.3