16. Consider the following conditional:

```
IF  XX = ZERO
    AND ZZ = 1
    OR XX NOT = ZERO
    AND ZZ NOT = 2
    PERFORM 900-FINISH
END-IF
```

Indicate whether `900-FINISH` will be performed if `XX` and `ZZ` contain the following:

	XX	ZZ		XX	ZZ
(a)	0	0	(e)	1	0
(b)	0	1	(f)	1	1
(c)	0	2	(g)	1	2
(d)	0	3	(h)	1	3

17. Write a program excerpt to determine whether a field called `FLDA`, with a `PIC 99`, contains an odd or even number.

Hint: You may use the `DIVIDE ... REMAINDER` for this, or some other technique.

18. Write a routine to enter a person's birth date and today's date. Determine if the person is under 25 years old. Note that if a person is born on 7/15/1980 and today's date is 7/14/2005, the person is not yet 25—he or she will be 25 on the next day. Display a message indicating if the person is under 25.

19. Write a routine to enter a date of last payment and today's date from a keyboard. `DISPLAY` a message indicating `'PAYMENT OVERDUE'` if more than 90 days have passed since the last payment.

20. Write a routine to enter a date of purchase and today's date. `DISPLAY` a message indicating the date payment is due, which is 30 days from the date of purchase.

Internet/Critical Thinking Questions

Your company is in the process of determining if it should maintain dates in Julian date mode (yyyyddd format), where yyyy is the year and ddd is the day of the year (1–365 or 366 in leap years). Some systems people prefer the standard mm/dd/yyyy format. Write a one-page summary of the advantages and disadvantages of using a Julian date. Use the Internet for source material if you need to.

DEBUGGING EXERCISES

1. Consider the following coding:

```
        .
        .
        .
    PERFORM UNTIL NO-MORE-RECORDS
        READ TRANS-FILE
            AT END
                MOVE 'NO ' TO ARE-THERE-MORE-RECORDS
            NOT AT END
                PERFORM 200-CALC-RTN
        END-READ
    END-PERFORM
        .
        .
        .
200-CALC-RTN.
    IF  AMT1 = 5400
        ADD AMT2 TO TOTAL
    ELSE
        ADD 1 TO ERR-CT
    WRITE OUT-REC FROM DETAIL-REC.
```

Under what conditions is a record written?
(*Hint:* The punctuation is more critical here than the indentations.)

2. The following coding will result in a syntax error. Explain why.

```
IF  AMT1 = AMT2
    ADD AMT3 TO TOTAL.
ELSE
    ADD AMT4 TO TOTAL.
```

3. Consider the following specifications:

```
01  REC-1.
    05  A          PIC X.
    05  B          PIC 9.
    05  C          PIC 9.
```

(a) The following coding will result in a syntax error. Explain why.

```
IF  A IS POSITIVE
    PERFORM 900-GO-TO-IT.
```

(b) Consider the following:

```
IF  A NOT EQUAL TO 6 OR
    A NOT EQUAL TO 7
    PERFORM 800-RTN-X.
```

Will a syntax error result? Explain your answer. Under what condition will `800-RTN-X` be performed?

(c) Suppose that `B` was not initialized and you included the following coding in the `PROCEDURE DIVISION`:

```
IF  B = 6
    PERFORM 500-RTN5.
```

Under what conditions, if any, will a syntax error occur? Under what conditions, if any, would a program interrupt occur?

PROGRAMMING ASSIGNMENTS

Because of the importance of conditional statements, an extended list of programming assignments has been included here. We recommend that you begin by planning your logic with a pseudocode and a hierarchy chart for each program before coding it.

1. Write a program for a rental car company that prints the amount owed by each customer. The amount owed depends on the miles driven, the number of days the car was rented, and the type of car rented. Toyotas rent at $26 per day and 18¢ per mile. Oldsmobiles rent at $32 per day and 22¢ per mile. Cadillacs rent for $43 per day and 28¢ per mile. The first 100 miles are free regardless of the car rented.

   The format of the input is as follows:

CUSTOMER Record Layout			
Field	**Size**	**Type**	**No. of Decimal Positions (if Numeric)**
Customer Last Name	20	Alphanumeric	
First Initial	1	Alphanumeric	
Type of Car	1	Alphanumeric: 1 = Toyota 2 = Oldsmobile 3 = Cadillac	
Miles Driven	5	Numeric	0
No. of Days Rented	3	Numeric	0

2. **Interactive Processing.** Do Programming Assignment 1 with interactive input and output. That is, write it so a clerk can use the computer to display the charges while a customer is waiting.
3. Write a program to list all employees who meet all of the following conditions:
   a. Annual salary is at least $20,000.
   b. Job classification code is 02.
   c. Territory number is 01.

   The problem definition is shown in Figure 8.4.
4. **Interactive Processing.** Do Programming Assignment 3 so that the output is displayed on the screen instead of printed. Also, enter the minimum salary, job classification, and territory interactively.

EVIEW QUESTIONS

True-False Questions

_____ 1. A PERFORM paragraph-name statement permanently transfers control to some other section of a program.

_____ 2. An in-line PERFORM UNTIL has a scope terminator.

_____ 3. PERFORM UNTIL X = Y and IF X = Y do the same thing.

_____ 4. It is generally better to test for ">" or ">=" a value than for "=" a value.

_____ 5. With a PERFORM UNTIL, if the condition is not met initially, then the instructions to be PERFORMed are not executed at all.

_____ 6. PERFORM 400-LOOP-RTN N TIMES is only valid if N is defined as numeric.

_____ 7. Using PERFORM 400-LOOP-RTN N TIMES, N should not be altered within 400-LOOP-RTN.

_____ 8. It is valid to say PERFORM 400-LOOP-RTN N TIMES, where N = 0.

_____ 9. The PERFORM and GO TO statements will cause identical branching.

_____ 10. With a PERFORM TIMES, you must establish a counter.

General Questions

1. Using a PERFORM statement with a TIMES option, write a routine to find N factorial, where N is the data item. Recall that N factorial = N × (N − 1) × (N − 2) × . . . × 1; e.g., 5 factorial = 5 × 4 × 3 × 2 × 1 = 120. Zero factorial is 1.
2. Rewrite the following routine using (a) a PERFORM statement with a TIMES option and (b) a PERFORM with a VARYING option:

```
    MOVE ZEROS TO COUNTER
    PERFORM 400-LOOP-RTN
        UNTIL COUNTER = 20.
    .
    .
    .
400-LOOP-RTN.
    DISPLAY 'QUANTITY?'
    ACCEPT QTY
    ADD QTY TO TOTAL
    ADD 1 TO COUNTER.
```

3. Write three routines, one with a PERFORM . . . TIMES, one with a PERFORM UNTIL . . ., and one with a PERFORM VARYING . . . to sum all odd integers from 1 to 1001, exclusive of 1001.
4. Write a routine to sum all even integers from 2 through 100 inclusive.
5. Write a routine to calculate the number of hours in a 365-day year using nested PERFORMs.
6. On January 1, you put $1.00 in the bank. On January 2, you put $2.00 in the bank. On January 3, you put $3.00 in the bank, and so on. Write a routine using PERFORM statements to DISPLAY the total amount of money you will have saved by the end of the year. Assume (1) that this is not a leap year and (2) that there is no interest earned on this money.
7. You earn $80,000 a year. At the end of the first week of the year, you save 1% of your weekly salary. At the end of the second week, you save 1.1% of your weekly salary. At the end of the third week, you save 1.2% of your weekly salary, and so on. Write a routine to DISPLAY the total amount of money you will have saved by the end of the year. Assume that there is no interest earned on the savings.

Internet/Critical Thinking Questions

1. Your company wishes to standardize the use of PERFORM statements. Would you recommend in-line PERFORMs or procedural PERFORMs as a standard? Which version of the PERFORM would you suggest as a standard: PERFORM . . . TIMES, PERFORM . . . UNTIL, or PERFORM . . . VARYING? Give reasons for your choices. Use Internet sites for help.
2. Use the Internet to determine why GO TO statements and their equivalents are avoided in most programming languages. Cite your sources.

DEBUGGING EXERCISES

1. Consider the following coding:

```
    PERFORM 400-ADD-RTN
        VARYING X FROM 1 BY 1 UNTIL X > 50.
    .
    .
    .
```

```
400-ADD-RTN.
    READ AMT-FILE
        AT END MOVE 'NO ' TO ARE-THERE-MORE-RECORDS
    END-READ
    ADD AMT TO TOTAL
    ADD 1 TO X.
```

(a) How many times is AMT added to TOTAL?
(b) Is the logic in the program excerpt correct? Explain your answer.
(c) What will happen if there are only 14 input records? Explain your answer.
(d) Correct the coding so that it adds amounts from 50 input records and prints an error message if there are fewer than 50 records.

2. Consider the following program excerpt:

```
    .
    .
    .
    PERFORM 200-CALC-RTN
        UNTIL NO-MORE-RECORDS
    .
    .
    .
200-CALC-RTN.
    READ SALES-FILE
        AT END MOVE 'NO ' TO ARE-THERE-MORE-RECORDS
    END-READ
    MOVE 0 TO COUNTER
    PERFORM 300-LOOP-RTN
        UNTIL COUNTER = 5
    MOVE TOTAL TO TOTAL-OUT
    MOVE TOTAL-REC TO PRINT-REC
    WRITE PRINT-REC.
300-LOOP-RTN.
    ADD AMT1 AMT2 GIVING AMT3
    MULTIPLY 1.08 BY AMT3 GIVING GROSS
    SUBTRACT DISCOUNT FROM GROSS
        GIVING TOTAL.
```

(a) This coding will result in a program interrupt. Indicate why. What changes should be made to correct the coding?
(b) Suppose COUNTER is initialized in WORKING-STORAGE with a VALUE of 0. Would it be correct to eliminate the MOVE 0 TO COUNTER instruction from 200-CALC-RTN? Explain your answer.
(c) Code the three arithmetic statements in 300-LOOP-RTN with a single COMPUTE statement.

PROGRAMMING ASSIGNMENTS

1. Write a program to produce a bonus report. See the problem definition in Figure 9.7.

Notes:

a. The payroll records have been sorted into ascending sequence by office number within territory number. There are three territories, two offices within each territory, and 10 employees within each office. We have, therefore, 60 records (3 × 2 × 10). Thus, all employees within office 01 within territory 01 will appear before employee records for office 02 within territory 01, and so on.
b. Only employees who were hired before 1994 are entitled to a 10% bonus.
c. Print the names of all employees and their bonuses. Print a separate page for each office within each territory. Ten employees will be printed on each page. Use a nested PERFORM to achieve page breaks for each office within each territory.

2. **Interactive Processing.** Write a program to display a temperature conversion table on a screen. Compute and print the Fahrenheit equivalents of all Celsius temperatures at 10-degree intervals from 0 to 150 degrees. The conversion formula is Celsius = 5/9 (Fahrenheit − 32).

Note: This program does not need an input data set.