II. General Questions

1. Consider student records that have the following format:
   1–15 Student Last Name
   16–25 Student First Name
   26–28 Student GPA (9V99)
   29–31 School (BUS = Business, ENG = Engineering, LAS = Liberal Arts)

   Records are in sequence by Student Name within School (BUS, ENG, LAS). A control break program is needed to print out the average GPA for each school.
   (a) Write the main module for this program.
   (b) Write the detail module for this program.
   (c) Write the control break module for this program.
   (d) Modify the modules above and add an end-of-job module that prints the average GPA for the entire university.
2. Which of the following is *not* true about control break processing?
   (a) At end-of-file a control break must be forced.
   (b) The final total is most efficiently calculated by adding the amounts from each input record.
   (c) A summary line is often written after the last control total is printed.
   (d) Each group's data is often begun on a new page.
   (e) It is useful to check for sequence errors.
3. To accumulate a final total, we can add the amount from each record or we can add the control totals. Which is more efficient?
4. A control break statement uses what statement to test for a change in control field?
5. After a group total is printed, what value must be moved into the field?

III. Internet/Critical Thinking Questions

This chapter on control break processing illustrates both detail and group printing. Using the Internet for source material, write a one-page analysis of the techniques that are commonly used for printing using COBOL. ***Hint:*** Perform a search on "COBOL" and "printing." Cite your sources.

DEBUGGING EXERCISES

Consider the following coding:

```
PROCEDURE DIVISION.
100-MAIN-MODULE.
    OPEN INPUT  TRANS-FILE
         OUTPUT PRINT-FILE
    PERFORM UNTIL THERE-ARE-NO-MORE-RECORDS
        READ TRANS-FILE
            AT END
                MOVE 'NO ' TO ARE-THERE-MORE-RECORDS
            NOT AT END
                MOVE ACCT-NO-IN TO WS-HOLD-ACCT
                PERFORM 200-DETAIL-RUN
        END-READ
    END-PERFORM
    CLOSE TRANS-FILE
          PRINT-FILE
    STOP RUN.
200-DETAIL-RTN.
    PERFORM 300-ADD-IT-UP
        UNTIL ACCT-NO IS NOT EQUAL TO WS-HOLD-ACCT OR
            THERE-ARE-NO-MORE-RECORDS
    MOVE WS-HOLD-ACCT TO ACCT-OUT
    MOVE WS-TOTAL TO TOTAL-OUT
    WRITE PRINT-REC FROM OUT-REC
    .
    .
300-ADD-IT-UP.
    ADD AMT TO WS-TOTAL
    READ TRANS-FILE
        AT END MOVE 'NO ' TO ARE-THERE-MORE-RECORDS
    END-READ.
```

1. Is the overall logical structure correct?
2. After executing the in-line `PERFORM UNTIL ...` in the main module, should there be a `PERFORM` to print the last control group? Explain your answer.
3. There are two instructions missing from `200-DETAIL-RTN` that will result in logic errors. Insert them.
4. Suppose `200-DETAIL-RTN` had a `READ` as its last instruction. How would this affect processing?
5. Suppose we omitted the `MOVE ACCT-NO-IN TO WS-HOLD-ACCT` statement from the main module. Would this have any substantial effect on the processing? Explain your answer.

PROGRAMMING ASSIGNMENTS

1. Write a program to print a sales total from disk records for each of five transaction days. The problem definition is shown in Figure 10.14.

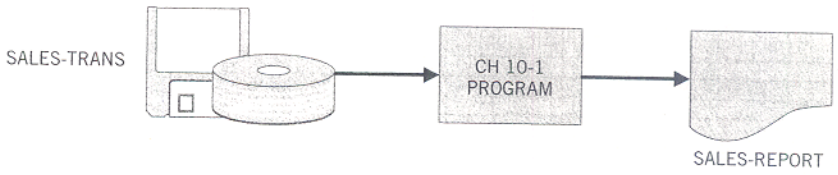
   Notes:

   a. There is a disk record for each sale made by a salesperson; thus there are an undetermined number of input records.
   b. Records are in sequence by day number, which ranges from 1 to 5 (Mon–Fri).

2. Write a program to print total salaries by territory number. The problem definition is shown in Figure 10.15. See, also, the notes on the next page.

Figure 10.14 Problem definition for Programming Assignment 1.

Systems Flowchart



SALES-TRANS Record Layout

Field	Size	Type	No. of Decimal Positions (if Numeric)
DAY-NO	1	Alphanumeric	
SALESPERSON-NO	3	Alphanumeric	
SALES-AMOUNT	5	Numeric	2

SALES-REPORT Printer Spacing Chart