

平成15年度 春期 FE 午後問題 COBOL

問7 次の COBOL プログラムの説明及びプログラムを読んで、設問に答えよ。

〔プログラムの説明〕

パソコンの修理状況を記録した修理状況ファイルから故障内容コード別に件数を集計し、故障内容コード別のレコード件数及びその比率を印字するプログラムである。

(1) 修理状況ファイルのレコード様式は、次のとおりである。

受付番号	受付日	故障内容コード	その他の情報
6 けた	8 けた	3 けた	26 けた

(2) 印字様式は、次のとおりである。

FAULT	CODE	COUNT	PERSENTAGE
XXX		Z,ZZ9	ZZ9.9
XXX		Z,ZZ9	ZZ9.9
XXX		Z,ZZ9	ZZ9.9
⋮		⋮	⋮
XXX		Z,ZZ9	ZZ9.9

故障内容コード別のレコード件数は、9,999 以下とする。

比率は、各故障内容コードの件数を修理状況ファイルの総レコード件数で割った百分率とする。小数第2位以下は、切り捨てる。

見出しの印字は、最初に1回行う。

〔プログラム〕

```

DATA DIVISION.
FILE SECTION.
FD REPAIR-FILE.
01 REPAIR-REC.
    02 ORDER-REP          PIC X(06).
    02 DATE-REP           PIC X(08).
    02 FAULT-REP          PIC X(03).
    02 EST-REP            PIC X(26).
FD PRINT-FILE.
01 PRINT-REC              PIC X(29).
SD SORT-FILE.
01 SORT-REC              PIC X(03).
WORKING-STORAGE SECTION.
01 END-ST                PIC 9(01).
01 COUNT-T               PIC 9(07).
01 FAULT-WRK             PIC X(03).
01 COUNT-WRK             PIC 9(04).
01 TOP-HEADER            PIC X(29) VALUE
    "FAULT CODE  COUNT  PERCENTAGE".
01 DATA-PRN.
    02 FAULT-PRN          PIC X(03).
    
```

```

02                        PIC X(09)
                           VALUE SPACE.
02 COUNT-PRN              PIC Z,ZZ9.
02                        PIC X(07)
                           VALUE SPACE.
02 PER-PRN                PIC ZZ9.9.
PROCEDURE DIVISION.
SORT-RTN.
    SORT SORT-FILE
      ON ASCENDING KEY SORT-REC
      INPUT PROCEDURE IS SELECT-RTN
      OUTPUT PROCEDURE IS TOTAL-RTN.
    STOP RUN.
SELECT-RTN.
    OPEN INPUT REPAIR-FILE.
    INITIALIZE END-ST COUNT-T.
    PERFORM UNTIL END-ST = 1
      READ REPAIR-FILE AT END
        MOVE 1 TO END-ST
      NOT AT END
        RELEASE SORT-REC FROM FAULT-REP
        COMPUTE COUNT-T = COUNT-T + 1
    END-READ.
    END-PERFORM.
    CLOSE REPAIR-FILE.
TOTAL-RTN.
    OPEN OUTPUT PRINT-FILE.
    WRITE PRINT-REC FROM TOP-HEADER AFTER PAGE.
    INITIALIZE END-ST.
    RETURN SORT-FILE AT END
      MOVE 1 TO END-ST
    NOT AT END
      PERFORM INIT-RTN
    END-RETURN.
    PERFORM UNTIL END-ST = 1
      RETURN SORT-FILE AT END
      PERFORM PRINT-RTN
      MOVE 1 TO END-ST
    NOT AT END
      IF 
        COMPUTE COUNT-WRK = COUNT-WRK + 1
      ELSE
        PERFORM PRINT-RTN
        
      END-IF
    END-RETURN.
    END-PERFORM.
    CLOSE PRINT-FILE.
PRINT-RTN.
    MOVE COUNT-WRK TO COUNT-PRN.
    .
    MOVE FAULT-WRK TO FAULT-PRN.
    .
INIT-RTN.
    MOVE 1 TO COUNT-WRK.
    .
    
```

設問 プログラム中の  に入れる正しい答えを、解答群の中から選べ。

a に関する解答群

- ア FAULT-WRK = FAULT-REP
- イ FAULT-WRK = SORT-REC
- ウ FAULT-WRK > FAULT-REP
- エ FAULT-WRK > SORT-REC
- オ FAULT-WRK NOT = FAULT-REP
- カ FAULT-WRK NOT = SORT-REC

b, e に関する解答群

- ア MOVE FAULT-REP TO FAULT-WRK
- イ MOVE FAULT-WRK TO FAULT-PRN
- ウ MOVE FAULT-WRK TO FAULT-REP
- エ MOVE FAULT-WRK TO SORT-REC
- オ MOVE SORT-REC TO FAULT-PRN
- カ MOVE SORT-REC TO FAULT-WRK
- キ PERFORM INIT-RTN
- ク PERFORM PRINT-RTN
- ケ PERFORM SELECT-RTN
- コ PERFORM TOTAL-RTN

c に関する解答群

- ア COMPUTE PER-PRN = 100 \* COUNT-T / COUNT-WRK
- イ COMPUTE PER-PRN = 100 \* COUNT-WRK / COUNT-T
- ウ COMPUTE PER-PRN = COUNT-T / COUNT-WRK
- エ COMPUTE PER-PRN = COUNT-T / COUNT-WRK / 100
- オ COMPUTE PER-PRN = COUNT-WRK / COUNT-T
- カ COMPUTE PER-PRN = COUNT-WRK / COUNT-T / 100
- キ MOVE COUNT-T TO PER-PRN
- ク MOVE COUNT-WRK TO PER-PRN

d に関する解答群

- ア WRITE PRINT-REC
- イ WRITE PRINT-REC AFTER 1
- ウ WRITE PRINT-REC FROM DATA-PRN AFTER 1
- エ WRITE PRINT-REC FROM TOP-HEADER AFTER 1

問 11 次の COBOL プログラムの説明及びプログラムを読んで、設問に答えよ。

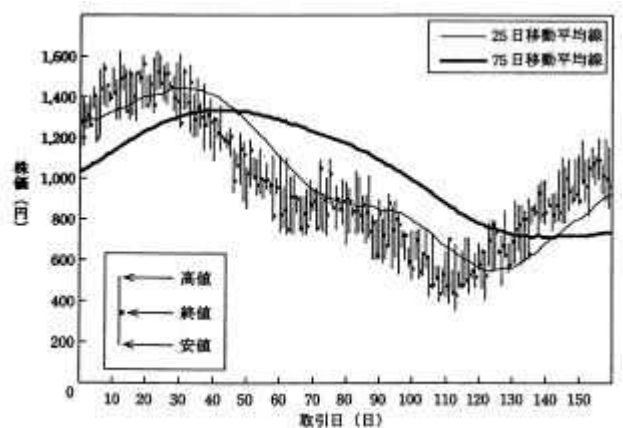
過去の株価情報を使い、次に説明する規則に従って株式の売買をシミュレートするプログラムである。ある銘柄の株式の日々の株価を格納したファイルを読み込み、この規則による売買の記録を出力ファイルに書き出し、最後に一連の売買の結果である損益を表示する。

〔プログラムの説明〕

(1) 株式の売買規則は次のとおりである。

25 日移動平均線と 75 日移動平均線を使う。ここで、 $n$  日移動平均線とは、時間軸を移動しながら過去  $n$  取引日間の株価の平均値を求めた折れ線グラフである。取引日  $t$  の株価を  $P_t$  とすると、取引日  $t$  における  $n$  日移動平均  $MA_t(n)$  は、次の式で求める。

$$MA_t(n) = \frac{P_t + P_{t-1} + \dots + P_{t-n+1}}{n}$$



移動平均の算出には、株価の終値（その日の最後に売買された値段）を使う。

25 日移動平均線が 75 日移動平均線の下から上へ交差したら、翌取引日に株式を買う。すなわち、 $s < t$  のとき、いったん  $MA_s(25) < MA_s(75)$  になってから、初めて  $MA_t(25) > MA_t(75)$  に変わったら、取引日  $t+1$  に株式を買う。

25 日移動平均線が 75 日移動平均線の上から下へ交差したら、翌取引日に株式を売る。すなわち、 $s < t$  のとき、いったん  $MA_s(25) > MA_s(75)$  になってから、初めて  $MA_t(25) < MA_t(75)$  に変わったら、取引日  $t+1$  に株式を売る。ただし、既に行った株式がなければ、売ることはできない。

株式の売買は、取引日  $t$  の高値（売買された最も高い値段）と安値（売買された最も低い値段）の平均値で売買できるものとする。

売買に関するその他の費用は、一切発生しないものとする。

株式の売買は、1 回あたり 1 株とする。

(2) 入力ファイル（INFILE）は、ある銘柄の株式について、75 日以上取引日の株価データが時系列に格納

**示現塾 プロジェクトマネージャ・テクニカルエンジニア(ネットワーク)など各種セミナーを開催中!!**

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス!!

された順ファイルである。レコード様式は、次のとおりである。

日付	始値	終値	高値	安値
8けた	9けた	9けた	9けた	9けた

(3) 出力ファイル(OUTFILE)は、入力ファイルの株価データに、(1)の売買規則を適用したときの売買の記録を格納する順ファイルである。レコード様式は、次のとおりである。

日付	フラグ	売買価格
8けた	1けた	10けた

出力ファイル(OUTFILE)に出力するレコードには、売買することを決定した日(取引日 t)の日付、フラグ及び売買価格を格納する。

フラグには、“売り”のとき“S”、“買い”のとき“B”を格納する。

売買価格には、小数点以下1けたで値を格納する。

(4) 一連の売買による損益は、小数点以下1けたの符号付き数値で表示する。

例：  

Total Revenue:	-114.5
----------------	--------

**{ プログラム }**

```

DATA DIVISION.
FILE SECTION.
FD INFILE.
01 I-MARKET-PRICE-REC.
   05 I-YYYYMMDD-MARKET PIC X(8).
   05 I-OPENING-PRICE PIC 9(9).
   05 I-CLOSING-PRICE PIC 9(9).
   05 I-HIGHEST-PRICE PIC 9(9).
   05 I-LOWEST-PRICE PIC 9(9).
FD OUTFILE.
01 O-EXCHANGE-REC.
   05 O-YYYYMMDD-EXCHANGE PIC X(8).
   05 O-ACTION PIC X.
   05 O-PRICE PIC 9(9)V9.
WORKING-STORAGE SECTION.
01 SERIES-SIZE PIC 9(4) VALUE 100.
01 RANGE-SHORT PIC 9(4) VALUE 25.
01 RANGE-LONG PIC 9(4) VALUE 75.
01 W-MARKET-PRICE-REC.
   05 W-YYYYMMDD-MARKET PIC X(8).
   05 W-OPENING-PRICE PIC 9(9).
   05 W-CLOSING-PRICE PIC 9(9).
   05 W-HIGHEST-PRICE PIC 9(9).
   05 W-LOWEST-PRICE PIC 9(9).
01 MARKET-PRICE-SERIES.
   05 TBL-MARKET-PRICE-REC OCCURS 100.
   10 TBL-YYYYMMDD-MARKET PIC X(8).
    
```

```

10 TBL-OPENING-PRICE PIC 9(9).
10 TBL-CLOSING-PRICE PIC 9(9).
10 TBL-HIGHEST-PRICE PIC 9(9).
10 TBL-LOWEST-PRICE PIC 9(9).
01 SERIES-TOP PIC 9(4).
01 TAIL-LONG PIC 9(4).
01 TAIL-SHORT PIC 9(4).
01 SUM-SHORT PIC 9(12).
01 SUM-LONG PIC 9(12).
01 MA-SHORT PIC 9(9)V9.
01 MA-LONG PIC 9(9)V9.
01 MA-COMPARE-CURRENT PIC X.
   88 LONG-LT-SHORT-CUR VALUE "S".
   88 SHORT-LT-LONG-CUR VALUE "L".
   88 SHORT-EQ-LONG-CUR VALUE "E".
01 MA-COMPARE-PREVIOUS PIC X.
   88 LONG-LT-SHORT-PRE VALUE "S".
   88 SHORT-LT-LONG-PRE VALUE "L".
   88 SHORT-EQ-LONG-PRE VALUE "E".
01 INFILE-EOF-FLG PIC X.
   88 INFILE-EOF VALUE "Y".
   88 INFILE-NOT-EOF VALUE "N".
01 TRADING-PRICE PIC 9(9)V9.
01 REVENUE PIC S9(9)V9.
01 REVENUE-ED PIC ++++++9.9.
01 INDX PIC 9(9).
01 TRADING-ACTION PIC S9.
   88 FLG-NO-TRADE VALUE 0.
   88 FLG-SELL VALUE 1.
   88 FLG-BUY VALUE -1.
01 STOCK-HOLDING PIC X.
   88 HAS-STOCK VALUE "Y".
   88 HAS-NO-STOCK VALUE "N".
PROCEDURE DIVISION.
MAIN-PARAGRAPH.
   OPEN INPUT INFILE OUTPUT OUTFILE.
   PERFORM INIT-MOVING-AVERAGE.
   IF MA-SHORT = MA-LONG THEN
      SET SHORT-EQ-LONG-CUR TO TRUE
   END-IF.
   PERFORM SET-CURRENT-MA-COMPARISON.
   MOVE ZERO TO REVENUE.
   SET HAS-NO-STOCK TO TRUE.
   SET FLG-NO-TRADE TO TRUE.
   SET INFILE-NOT-EOF TO TRUE.
   PERFORM UNTIL INFILE-EOF
      IF FLG-BUY OR FLG-SELL THEN
         PERFORM TRADE-STOCK-AND-MAKE-RECORD
      END-IF
      READ INFILE INTO W-MARKET-PRICE-REC
      AT END
         SET INFILE-EOF TO TRUE
      NOT AT END
         PERFORM CALC-NEXT-MOVING-AVERAGE
         PERFORM DECIDE-TRADING
      END-READ
   END-PERFORM.
   MOVE REVENUE TO REVENUE-ED.
   DISPLAY "Total Revenue: " REVENUE-ED.
   CLOSE INFILE OUTFILE.
   STOP RUN.
    
```

示現塾 プロジェクトマネージャ・テクニカルエンジニア(ネットワーク)など各種セミナーを開催中！！

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス！！

```

INIT-MOVING-AVERAGE.
  PERFORM VARYING SERIES-TOP FROM 1 BY 1
    UNTIL RANGE-LONG < SERIES-TOP
  READ INFILE INTO
    TBL-MARKET-PRICE-REC(SERIES-TOP)
  END-PERFORM.
  COMPUTE TAIL-SHORT = SERIES-TOP -
    RANGE-SHORT.
  COMPUTE TAIL-LONG = SERIES-TOP -
    RANGE-LONG.
  SUBTRACT 1 FROM SERIES-TOP.
  MOVE ZERO TO SUM-SHORT.
  PERFORM VARYING INDX
     a
  ADD TBL-CLOSING-PRICE(INDX) TO SUM-SHORT
  END-PERFORM.
  MOVE SUM-SHORT TO SUM-LONG.
  PERFORM VARYING INDX
     b
  ADD TBL-CLOSING-PRICE(INDX) TO SUM-LONG
  END-PERFORM.
  COMPUTE MA-SHORT ROUNDED = SUM-SHORT /
    RANGE-SHORT.
  COMPUTE MA-LONG ROUNDED = SUM-LONG /
    RANGE-LONG.
SET-CURRENT-MA-COMPARISON.
  EVALUATE TRUE
    WHEN MA-SHORT = MA-LONG
      CONTINUE
    WHEN MA-SHORT < MA-LONG
      SET SHORT-LT-LONG-CUR TO TRUE
    WHEN MA-LONG < MA-SHORT
      SET LONG-LT-SHORT-CUR TO TRUE
  END-EVALUATE.
TRADE-STOCK-AND-MAKE-RECORD.
  EVALUATE TRUE
    WHEN FLG-BUY
      MOVE "B" TO O-ACTION
      SET HAS-STOCK TO TRUE
    WHEN FLG-SELL AND  c
      MOVE "S" TO O-ACTION
      SET HAS-NO-STOCK TO TRUE
  END-EVALUATE.
  IF O-ACTION = "B" OR O-ACTION = "S"
    THEN
      COMPUTE TRADING-PRICE ROUNDED
        = ( TBL-HIGHEST-PRICE(SERIES-TOP)
          + TBL-LOWEST-PRICE(SERIES-TOP) )
          / 2
      COMPUTE REVENUE = REVENUE + TRADING-PRICE
        *  d
      MOVE TBL-YYYYMMDD-MARKET(SERIES-TOP) TO
        O-YYYYMMDD-EXCHANGE
      MOVE TRADING-PRICE TO O-PRICE
      WRITE O-EXCHANGE-REC
    END-IF.
CALC-NEXT-MOVING-AVERAGE.
  ADD W-CLOSING-PRICE TO SUM-SHORT SUM-LONG.
  SUBTRACT TBL-CLOSING-PRICE(TAIL-SHORT)
    FROM SUM-SHORT.
  SUBTRACT TBL-CLOSING-PRICE(TAIL-LONG)

```

```

    FROM SUM-LONG.
  COMPUTE MA-SHORT ROUNDED = SUM-SHORT
    / RANGE-SHORT.
  COMPUTE MA-LONG ROUNDED = SUM-LONG
    / RANGE-LONG.
  ADD 1 TO SERIES-TOP TAIL-SHORT TAIL-LONG.
  EVALUATE TRUE
    WHEN SERIES-SIZE < SERIES-TOP
      MOVE 1 TO SERIES-TOP
    WHEN SERIES-SIZE < TAIL-SHORT
      MOVE 1 TO TAIL-SHORT
    WHEN SERIES-SIZE < TAIL-LONG
      MOVE 1 TO TAIL-LONG
  END-EVALUATE.
  MOVE W-MARKET-PRICE-REC TO
    TBL-MARKET-PRICE-REC(SERIES-TOP).
DECIDE-TRADING.
  MOVE MA-COMPARE-CURRENT TO
    MA-COMPARE-PREVIOUS.
  PERFORM SET-CURRENT-MA-COMPARISON.
  SET FLG-NO-TRADE TO TRUE.
  EVALUATE TRUE
    WHEN  e AND LONG-LT-SHORT-CUR
      SET FLG-BUY TO TRUE
    WHEN LONG-LT-SHORT-PRE AND
      SHORT-LT-LONG-CUR
      SET FLG-SELL TO TRUE
  END-EVALUATE.

```

設問 プログラム中の  に入れる正しい答えを、  
解答群の中から選べ。

a, b に関する解答群

- ア FROM 1 BY 1 UNTIL SERIES-SIZE < INDX
- イ FROM SERIES-TOP - 1 BY -1 UNTIL INDX  
< TAIL-SHORT
- ウ FROM SERIES-TOP BY -1 UNTIL INDX  
<= TAIL-SHORT
- エ FROM TAIL-LONG BY 1 UNTIL TAIL-SHORT  
<= INDX
- オ FROM TAIL-SHORT BY 1 UNTIL SERIES-TOP  
< INDX

c ~ e に関する解答群

- ア FLG-BUY
- イ FLG-SELL
- ウ HAS-STOCK
- エ LONG-LT-SHORT-CUR
- オ MA-COMPARE-CURRENT
- カ SHORT-LT-LONG-PRE
- キ TRADING-ACTION

示現塾 プロジェクトマネージャ・テクニカルエンジニア(ネットワーク)など各種セミナーを開催中!!

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス!!

平成15年度 秋期 FE 午後問題 COBOL

問7 次の COBOL プログラムの説明及びプログラムを読んで、設問に答えよ。

〔プログラムの説明〕

予約状況ファイルと料金ファイルから、パラメタで与えられた検索条件を満たす宿泊施設の空室を検索して、宿泊料金の安い順に並べた空室情報を返す副プログラムである。

(1) 予約状況ファイル BOOKING-FILE のレコード様式は、次のとおりである。

施設コード	部屋名称	部屋タイプ	予約状況
3けた	10けた	2けた	1けた×366

施設コードは宿泊施設を表し、各施設には複数の部屋があるものとする。

予約状況は、366個の要素からなる1次元の表であり、プログラムを実行する翌日から1年間分の予約状況を表している。

要素の値が、0のときは“予約なし”を示し、1のときは“予約済”を示す。予約状況ファイルは毎日更新され、プログラム実行時には、それまでの予約状況が正しく反映されているものとする。

例 プログラムを実行する日が2003年10月19日のときは、2003年10月20日～2004年10月19日の予約状況が格納されている。

(2) 料金ファイル CHARGE-FILE は部屋タイプとする索引ファイルであり、レコード様式は次のとおりである。

部屋タイプ	収容情報1		収容情報2		収容情報4	
	宿泊人数	宿泊料金	宿泊人数	宿泊料金	宿泊人数	宿泊料金
2けた	2けた	6けた	2けた	6けた	2けた	6けた

宿泊人数：1部屋に泊まる人数を示す。

宿泊料金：部屋の1人当たりの1泊料金を示す。宿泊人数の違いによって4通り用意されている。

予約状況ファイルに格納されている部屋タイプに該当するレコードは、必ず存在する。

(3) 検索条件 SEARCH-INFO の形式は、次のとおりである。検索条件で与えられるデータは正しいものとする。

施設コード	宿泊開始日	宿泊日数	宿泊人数	1泊の予算	
				下限	上限
3けた	8けた	2けた	2けた	6けた	6けた

施設コード：3けたの数字である。“000”のときは施設の指定がないことを表す。

宿泊開始日：4けたの西暦年、2けたの月、2けたの日で構成されている。プログラムを実行

する翌日から1年以内の日付とする。

例 2003年10月20日ならば“20031020”となる。

宿泊日数：2けたの0でない数字である。宿泊開始日から連続して泊まる日数を表す。連続して泊まる日を含めて、プログラムを実行する翌日から1年間に収まる日数とする。

例 1泊2日のときは、“01”となる。

宿泊人数：2けたの0でない数字である。

1泊の予算：1人当たりの1泊の予算であり、下限と上限を示す1組の6けたの数字である。下限と上限がともに“000000”のときは指定なしを表す。

例 10,000～15,000円の部屋を検索するときは、下限として“010000”，上限として“015000”を指定する。

(4) 検索結果 SEARCH-RESULT の形式は、次のとおりである。

検索件数	空室情報1			空室情報100		
	施設コード	部屋名称	宿泊料金	施設コード	部屋名称	宿泊料金
3けた	3けた	10けた	6けた	3けた	10けた	6けた

空室情報は100個の要素からなる1次元の表であり、検索結果は100件以下とする。検索結果が100件未満のとき、残りの要素は、空白で埋められている。

空室情報は、宿泊料金の安い順に並べる。

(5) 副プログラム GETDAYS を使用する。GETDAYS は、与えられた日付データを、プログラムを実行する翌日からの通算日に変換する。GETDAYS の呼び出し方法は、次のとおりである。

CALL "GETDAYS" USING 日付 通算日

〔プログラム〕

```

DATA          DIVISION.
FILE          SECTION.

FD BOOKING-FILE.
01 BOOKING-REC.
   03 BCODE          PIC 9(03).
   03 BNAME          PIC X(10).
   03 BTYPE          PIC X(02).
   03 BOOKING-INFO  PIC 9(01) OCCURS 366.

FD CHARGE-FILE.
01 CHARGE-REC.
   03 CTYPE          PIC X(02).
   03 ACCOMMODATION-INFO OCCURS 4.
   05 GUEST-NO      PIC 9(02).
   05 CHARGE        PIC 9(06).
    
```

示現塾 プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス！！

```

SD VACANCY-FILE.
01 VACANCY-REC.
   03 VCODE          PIC 9(03).
   03 VNAME          PIC X(10).
   03 VCHARGE        PIC 9(06).
WORKING-STORAGE SECTION.
01 EOF              PIC X(01).
01 BOOKING-FLAG    PIC 9(01).
01 DAYS            PIC 9(03).
01 MATCH          PIC 9(01).
01 I              PIC 9(03).
01 J              PIC 9(01).
LINKAGE SECTION.
01 SEARCH-INFO.
   03 SCODE         PIC 9(03).
   03 CHECKIN-DATE PIC 9(08).
   03 STAY-DAYS     PIC 9(02).
   03 SGUEST-NO    PIC 9(02).
   03 SCHARGE-RANGE.
       05 LOWER-LIMIT PIC 9(06).
       05 UPPER-LIMIT PIC 9(06).
01 SEARCH-RESULT.
   03 SEARCH-COUNT PIC 9(03).
   03 VACANCY-INFO PIC X(19) OCCURS 100.
PROCEDURE DIVISION USING SEARCH-INFO
                           SEARCH-RESULT.

MAIN-PROC.
  SORT VACANCY-FILE
    [ a ]
  INPUT PROCEDURE SEARCH-PROC
  OUTPUT PROCEDURE RESULT-PROC.
  EXIT PROGRAM.
SEARCH-PROC.
  OPEN INPUT BOOKING-FILE CHARGE-FILE.
  MOVE SPACE TO EOF.
  READ BOOKING-FILE AT END
  MOVE "E" TO EOF
  END-READ.
  PERFORM UNTIL EOF = "E"
    IF (SCODE = ZERO OR
        SCODE = BCODE) THEN
      CALL "GETDAYS" USING CHECKIN-DATE DAYS
      MOVE 0 TO BOOKING-FLAG
      PERFORM [ b ]
      IF BOOKING-INFO(I) = 1 THEN
        MOVE 1 TO BOOKING-FLAG
      END-IF
    END-PERFORM
    IF [ c ] THEN
      MOVE BTYPE TO CTYPE
      READ CHARGE-FILE
      MOVE 0 TO MATCH
      PERFORM VARYING J FROM 1 BY 1
        UNTIL J > 4 OR MATCH = 1
      IF [ d ] THEN
        MOVE BCODE TO VCODE
        MOVE BNAME TO VNAME
        MOVE CHARGE(J) TO VCHARGE
        RELEASE VACANCY-REC
        MOVE 1 TO MATCH
      END-IF
    END-IF
  END-IF

```

```

END-PERFORM
END-IF
END-IF
READ BOOKING-FILE AT END
MOVE "E" TO EOF
END-READ
END-PERFORM.
CLOSE BOOKING-FILE CHARGE-FILE.
RESULT-PROC.
  INITIALIZE SEARCH-RESULT.
  MOVE SPACE TO EOF.
  RETURN VACANCY-FILE AT END
  MOVE "E" TO EOF
  END-RETURN.
  PERFORM UNTIL EOF = "E"
    COMPUTE SEARCH-COUNT = SEARCH-COUNT + 1
    MOVE VACANCY-REC TO
      VACANCY-INFO(SEARCH-COUNT)
  RETURN VACANCY-FILE AT END
  MOVE "E" TO EOF
  END-RETURN
  END-PERFORM.

```

設問 プログラム中の [ ] に入れる正しい答えを、  
解答群の中から選べ。

a に関する解答群

- ア OCCURS KEY VCHARGE
- イ ON ASCENDING KEY VCHARGE
- ウ ON DESCENDING KEY VCHARGE
- エ SORT KEY VCHARGE
- オ USING KEY VCHARGE

b に関する解答群

- ア VARYING I FROM 1 BY 1 UNTIL I > DAYS
- イ VARYING I FROM 1 BY 1 UNTIL I > DAYS  
OR BOOKING-FLAG = 1
- ウ VARYING I FROM 1 BY 1 UNTIL I >= STAY-DAYS
- エ VARYING I FROM 1 BY 1 UNTIL I >= STAY-DAYS  
OR BOOKING-FLAG = 1
- オ VARYING I FROM DAYS BY 1 UNTIL I >= DAYS  
+ STAY-DAYS OR BOOKING-FLAG = 1

c に関する解答群

- ア BOOKING-FLAG = 0
- イ BOOKING-FLAG = 1
- ウ BOOKING-INFO(I) = 0
- エ BOOKING-INFO(I) = 1
- オ SCODE = ZERO
- カ SCODE NOT = ZERO

示現塾 プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス！！

d に関する解答群

- ア (GUEST-NO(J) = SGUEST-NO AND  
(SCHARGE-RANGE = ZERO OR  
(CHARGE(J) >= LOWER-LIMIT AND CHARGE(J)  
<= UPPER-LIMIT)))
- イ (GUEST-NO(J) = SGUEST-NO AND  
(SCHARGE-RANGE NOT = ZERO AND  
(CHARGE(J) < LOWER-LIMIT OR CHARGE(J)  
> UPPER-LIMIT)))
- ウ (GUEST-NO(J) = SGUEST-NO OR  
(SCHARGE-RANGE = ZERO OR  
(CHARGE(J) >= LOWER-LIMIT AND CHARGE(J)  
<= UPPER-LIMIT)))
- エ NOT (GUEST-NO(J) NOT = SGUEST-NO AND  
(SCHARGE-RANGE = ZERO OR  
(CHARGE(J) >= LOWER-LIMIT AND CHARGE(J)  
<= UPPER-LIMIT)))
- オ NOT (GUEST-NO(J) NOT = SGUEST-NO AND  
(SCHARGE-RANGE NOT = ZERO AND  
(CHARGE(J) < LOWER-LIMIT OR CHARGE(J)  
> UPPER-LIMIT)))

問 11 次の COBOL プログラムの説明及びプログラムを読んで、設問に答えよ。

〔プログラムの説明〕

航空便を使った旅程の中で、出発地の空港から目的地の空港までの飛行時間と乗継時間を求めるプログラムである。出発する空港と日時及び到着する空港と日時を一組にしたレコードを、利用する航空便の数だけファイルに格納している。格納されている日時はそれぞれの空港の現地日時である。各空港の現地時間の世界標準時からの時差を格納した索引ファイルを参照しながら、一連の航空便の飛行時間、飛行時間合計及び乗継時間合計を求めて印字する。この問題では、夏時間は考慮しない。

(1) 飛行予定情報ファイル itinerary-file は、次のレコード様式の順ファイルである。

便名 6 けた	出発空港と日時				到着空港と日時			
	空港 コード 3 けた	日付 8 けた	時刻		空港 コード 3 けた	日付 8 けた	時刻	
			時 2 けた	分 2 けた			時 2 けた	分 2 けた

プログラムでは、“出発空港と日時”を繰返し項目の 1 番目の要素で、“到着空港と日時”を繰返し項目の 2 番目の要素で表現している。

レコードは、対象となる旅程で利用する航空便の順番に格納されている。あるレコードの到着時刻から次のレコードの出発時刻までが乗継時間である。一つの航空便で直接目的地に到達する場合は、レコードは一つであり、乗継ぎはない。

便名は 6 けた以内の英数字項目であり、左詰めに格納されている。

空港コードは、英字 3 文字のコードであり、コードは重複しない。

日付は、4 けたの西暦年 YYYY、2 けたの月 MM 及び 2 けたの日 DD を、YYYYMMDD の形の 10 進数として格納している。

(2) 空港と現地時間の世界標準時からの時差を格納するファイル airports-file のレコード様式は、次のとおりである。

空港コード 3 けた	世界標準時からの時差	
	時 3 けた	分 2 けた

3 文字の空港コードをキーとする索引ファイルである。

世界標準時からの時差の“時”は、符号付き数字項目であり、1 けたの符号に続く 2 けたの数字で表現されている。“分”は、符号なし数字項目である。

飛行予定情報ファイルに格納されている空港コードは、必ずファイル中に存在する。

(3) 一連の航空便の飛行時間、飛行時間合計及び乗継時間合計の印字例を、図 1 に示す。このときのプログラムへの入力は、図 2 の飛行予定情報ファイルと図 3 の空港と現地時間の世界標準時からの時差を格納するファイルである。各航空便について、便名、出発空港コード、出発日付、出発時刻、到着空港コード、到着日付、到着時刻、飛行時間を印字し、最後に飛行時間合計及び乗継時間合計を印字する。飛行時間合計と乗継時間合計は、それぞれ 100 時間未満とする。

```
XY1234 ABC 20030927 1715 to DEF 20030927 1030, flight time = 10:15
YX4567 DEF 20030927 1200 to GHI 20030927 1530, flight time = 02:30
Total flight time = 12:45
Total transit time = 01:30
```

図 1 印字例

```
XY1234ABC200309271715DEF200309271030
YX4567DEF200309271200GHI200309271530
```

図 2 飛行予定情報ファイルの例

```
ABC+0700
DEF-1000
GHI-0900
JFK-0500
LAX-0800
NRT+0900
ORD-0600
SAN-0800
```

図 3 空港と現地時間の世界標準時からの時差を格納するファイルの例

示現塾 プロジェクトマネージャ・テクニカルエンジニア(ネットワーク)など各種セミナーを開催中!!

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス!!

- (4) プログラム中で使われている組込み関数 INTEGER-OF-DATE は、飛行予定情報ファイルの日付の形式の整数を引数として受け取り、グレゴリオ暦で1600年12月31日からその日までの総日数を返す。

[プログラム]

```

DATA DIVISION.
FILE SECTION.
FD airports-file.
01 airport-timediff-rec.
   05 3-letter-code          PIC X(3).
   05 local-timediff.
      10 local-hh-diff       PIC S9(2)
                                LEADING SEPARATE.
      10 local-mm-diff       PIC 9(2).
FD itinerary-file.
01 flight-schedule.
   05 flight-number          PIC X(6).
   05 flight-time-and-location occurs 2.
      10 airport-of-call     PIC X(3).
      10 arrv-dprt-date-grp.
         15 arrv-dprt-date   PIC 9(8).
         10 arrv-dprt-time.
            15 arrv-dprt-hh   PIC 9(2).
            15 arrv-dprt-mm   PIC 9(2).
WORKING-STORAGE SECTION.
01 departure                 PIC 9 VALUE 1.
01 arrival                   PIC 9 VALUE 2.
01 this-airport              PIC 9.
01 .
   05 normalized-minutes     PIC 9(9) OCCURS 2.
01 arrival-normalized-minutes PIC 9(9).
01 time-in-minutes           PIC 9(9).
01 time-diff-in-minutes      PIC S9(9).
01 flight-minutes            PIC S9(9).
01 transit-minutes           PIC S9(9).
01 accumulated-flight-minutes PIC 9(9).
01 accumulated-transit-minutes PIC 9(9).
01 elapsed-time-hhmm.
   05 elapsed-hours          PIC 9(2).
   05                        PIC X VALUE ":".
   05 elapsed-minutes        PIC 9(2).
01 end-of-itinerary-file     PIC X.
PROCEDURE DIVISION.
main-paragraph.
  OPEN INPUT airports-file itinerary-file.
  MOVE "n" TO end-of-itinerary-file.
  INITIALIZE accumulated-flight-minutes
             accumulated-transit-minutes
             arrival-normalized-minutes
             transit-minutes.
  PERFORM UNTIL end-of-itinerary-file = "y"
    READ itinerary-file AT END
      MOVE "y" TO end-of-itinerary-file
      NOT AT END
        PERFORM calculate-flight-time
        ADD transit-minutes TO
            accumulated-transit-minutes
        ADD flight-minutes TO

```

```

            accumulated-flight-minutes
            PERFORM display-flight-time
            END-READ
            END-PERFORM.
            PERFORM display-accumulated-time.
            CLOSE airports-file itinerary-file.
            STOP RUN.
calculate-flight-time.
  MOVE arrival TO this-airport.
  PERFORM normalize-time.
  MOVE departure TO this-airport.
  PERFORM normalize-time.
  IF [a] THEN
    COMPUTE transit-minutes
      = [b]
  END-IF.
  COMPUTE flight-minutes
    = [c].
  MOVE normalized-minutes(arrival)
    TO arrival-normalized-minutes.
normalize-time.
  MOVE [d] TO
                                3-letter-code.
  READ airports-file.
  IF 0 <= local-hh-diff THEN
    COMPUTE time-diff-in-minutes
      = local-hh-diff * 60 + local-mm-diff
  ELSE
    COMPUTE time-diff-in-minutes
      = local-hh-diff * 60 - local-mm-diff
  END-IF.
  COMPUTE [e]
    = (FUNCTION INTEGER-OF-DATE(arrv-
                                dprt-date(this-airport))
      * 24 + arrv-dprt-hh(this-airport))
      * 60 + arrv-dprt-mm(this-airport) -
                                time-diff-in-minutes.
display-flight-time.
  MOVE flight-minutes TO time-in-minutes.
  PERFORM transform-minutes-to-hhmm.
  DISPLAY flight-number          " "
             airport-of-call(departure) " "
             arrv-dprt-date-grp(departure) " "
             arrv-dprt-time(departure) " to "
             airport-of-call(arrival) " "
             arrv-dprt-date-grp(arrival) " "
             arrv-dprt-time(arrival) " ,
                                flight time = "
             elapsed-time-hhmm.
display-accumulated-time.
  MOVE accumulated-flight-minutes TO
                                time-in-minutes.
  PERFORM transform-minutes-to-hhmm.
  DISPLAY "Total flight time = "
                                elapsed-time-hhmm.
  MOVE accumulated-transit-minutes TO
                                time-in-minutes.
  PERFORM transform-minutes-to-hhmm.
  DISPLAY "Total transit time = "
                                elapsed-time-hhmm.
transform-minutes-to-hhmm.

```



```
DIVIDE time-in-minutes BY 60
      GIVING elapsed-hours
      REMAINDER elapsed-minutes.
```

設問 プログラム中の  に入れる正しい答えを、  
解答群の中から選べ。

a に関する解答群

- ア 0 < arrival-normalized-minutes
- イ 0 < normalized-minutes(arrival)
- ウ 0 < normalized-minutes(departure)
- エ 0 = arrival-normalized-minutes

b, c に関する解答群

- ア arrival-normalized-minutes  
- normalized-minutes(departure)
- イ normalized-minutes(arrival)  
- normalized-minutes(departure)
- ウ normalized-minutes(departure)  
- arrival-normalized-minutes
- エ normalized-minutes(departure)  
- normalized-minutes(arrival)
- オ normalized-minutes(this-airport)  
- normalized-minutes(departure)

d, e に関する解答群

- ア airport-of-call(arrival)
- イ airport-of-call(this-airport)
- ウ normalized-minutes
- エ normalized-minutes(arrival)
- オ normalized-minutes(this-airport)
- カ this-airport

平成 15 年度 春期 F E 午後解答 COBOL

問 7

設問 a - イ      b - キ      c - イ      d - ウ      e - カ

問 11

設問 a - オ      b - エ      c - ウ      d - キ      e - カ

平成 15 年度 秋期 F E 午後解答 COBOL

問 7

設問 a - イ      b - オ      c - ア      d - ア

問 11

設問 a - ア      b - ウ      c - イ      d - イ      e - オ