FILE Simulation Project

FILE Simulation Project

This project Creates New Master File, Updates and stores data(transactions) on the File from the Intra-Day Transaction File.

Overview

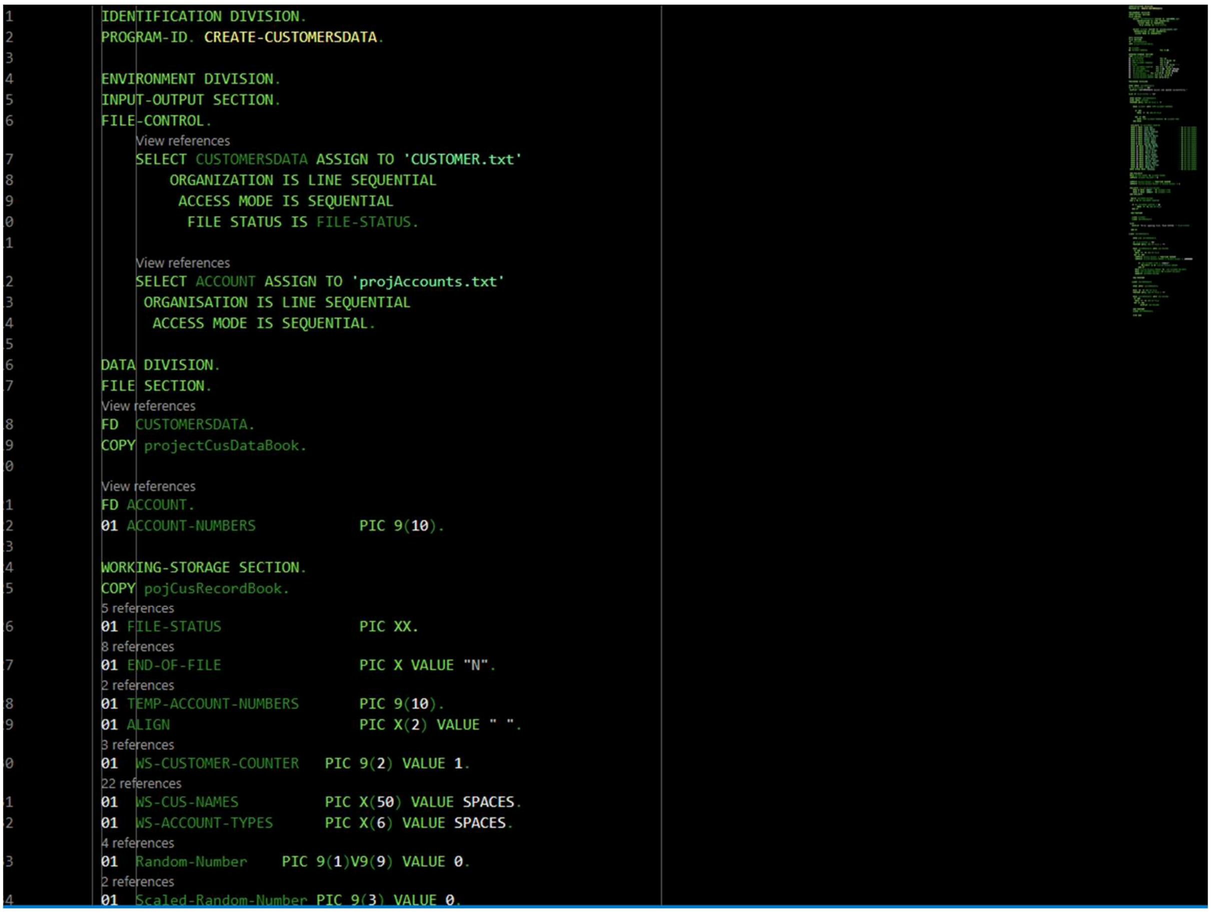
A brief documentation of a bank transaction system (Project). It entails designing and implementing a COBOL batch program to process transactions against a master file, generate reports, and update a history file. The program will ensure data consistency and accuracy by using header and trailer records for validation and will include date of the transaction and hash total checks

STRUCTURE

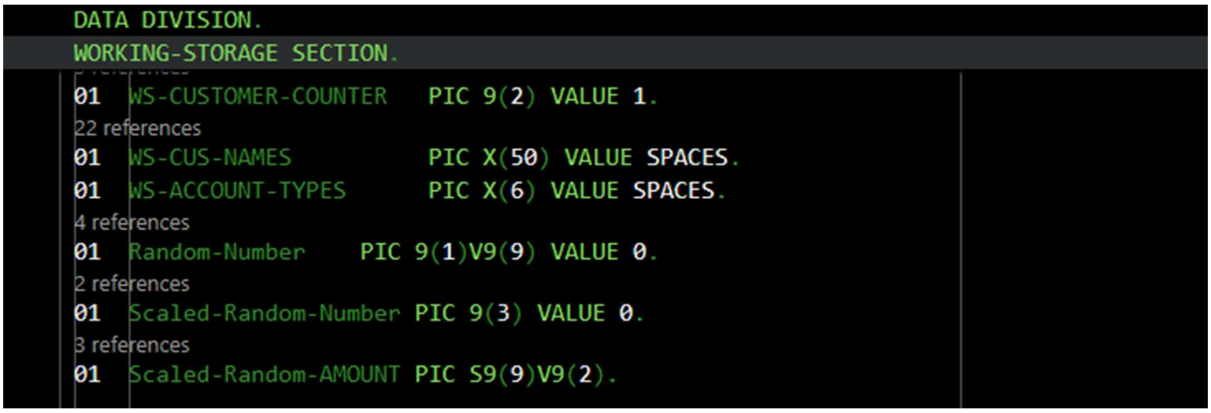
- **Master File:**
  - Fields: Account Number, Customer ID, Customer Name, Account Type, Balance.
- **Transaction File:**
  - Records: Header, Detail, and Trailer.
  - Fields: Transaction Date, Account Number, Customer ID, Account Type, Account Balance.
- **History File:**
  - Updated with all processed transactions details.
- **Validation Checks:**
  - Transaction date must match the current date.
  - Hash totals in the trailer must match the processed record count.
- **Reports:**
  - Display processed transactions and updated account balances.

The code and its components:

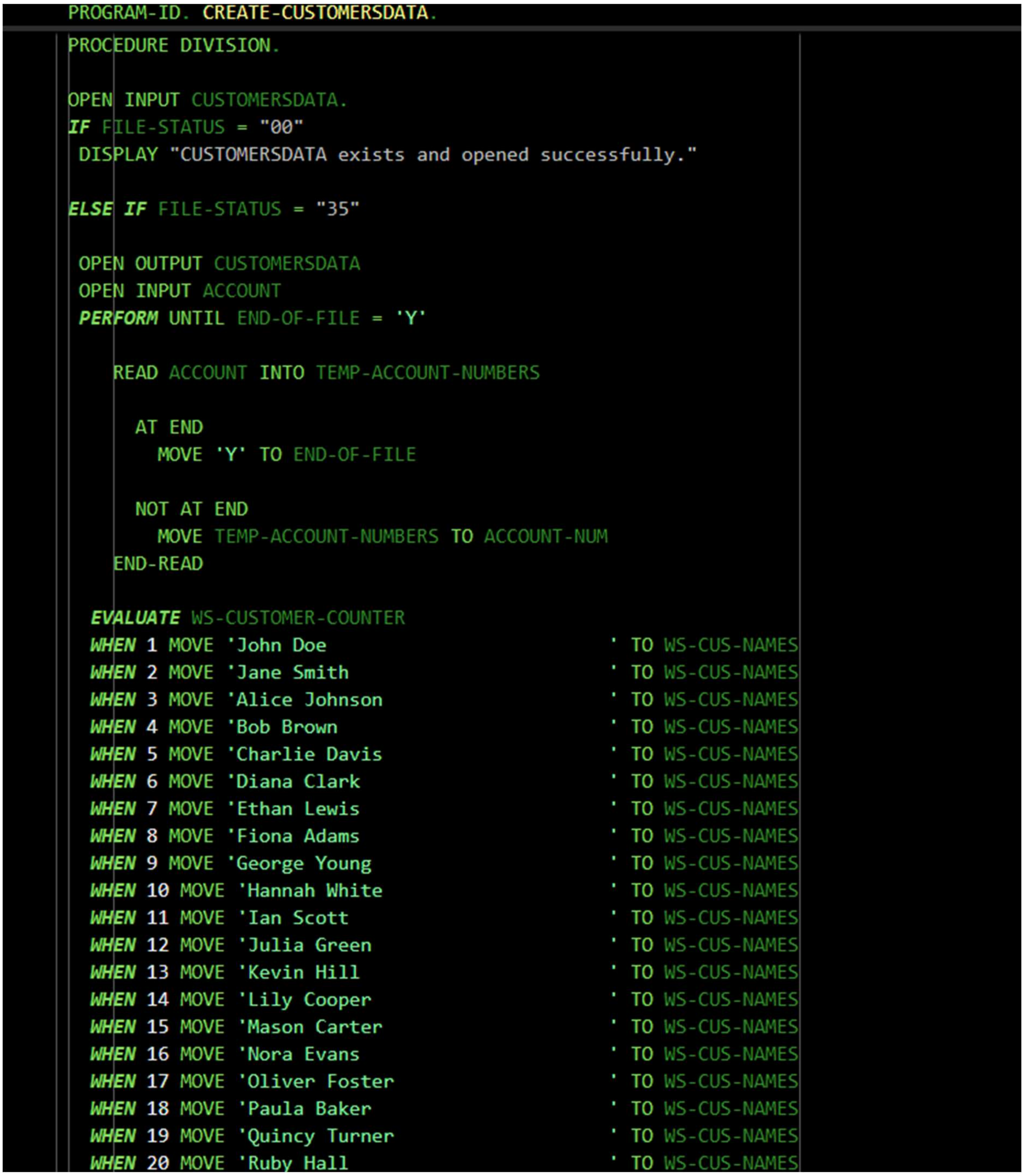
1.Customer generation



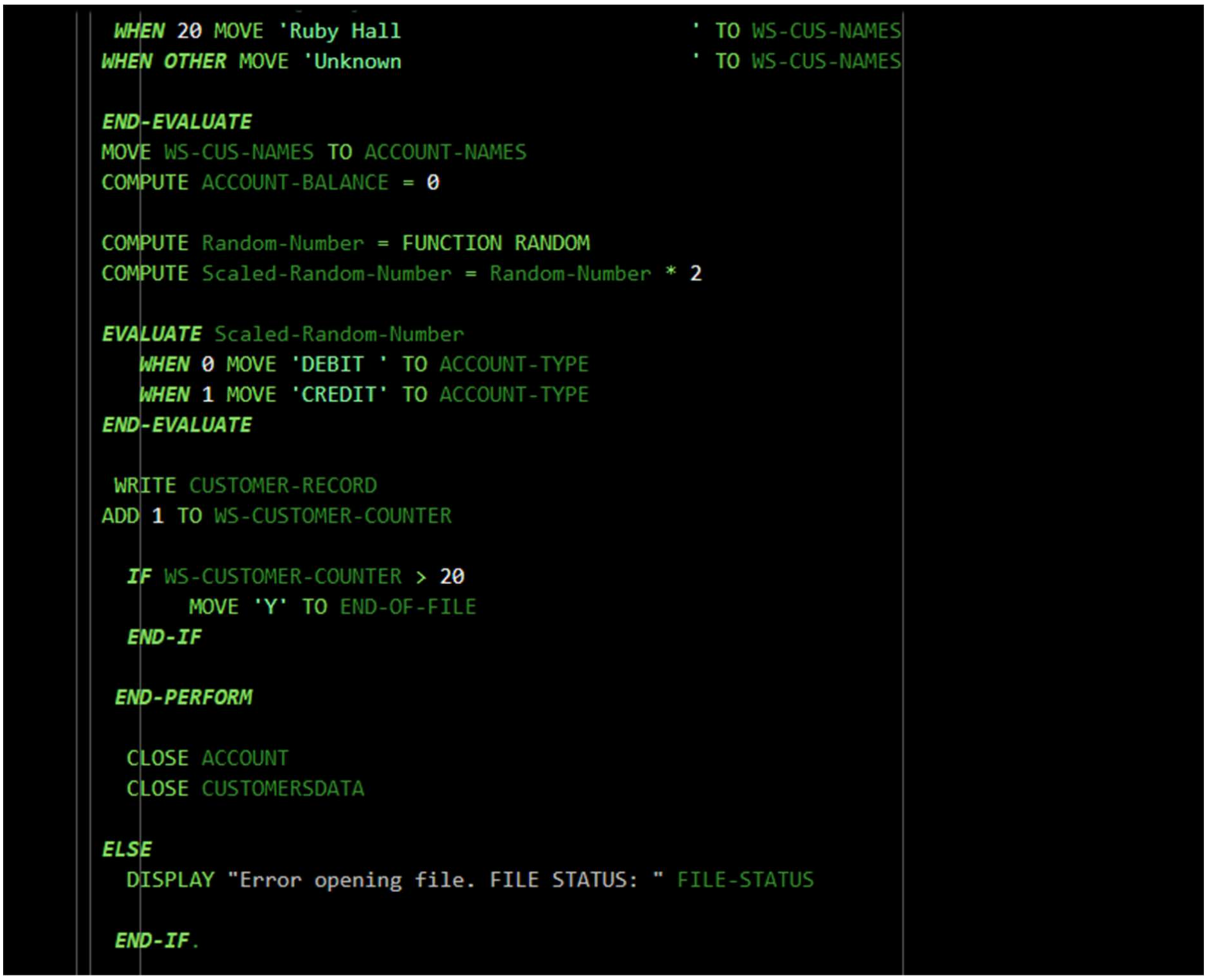
```
IDENTIFICATION DIVISION.
PROGRAM-ID. CREATE-CUSTOMERSDATA.

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT CUSTOMERSDATA ASSIGN TO 'CUSTOMER.txt'
        ORGANIZATION IS LINE SEQUENTIAL
         ACCESS MODE IS SEQUENTIAL
          FILE STATUS IS FILE-STATUS.

    SELECT ACCOUNT ASSIGN TO 'projAccounts.txt'
     ORGANISATION IS LINE SEQUENTIAL
      ACCESS MODE IS SEQUENTIAL.

DATA DIVISION.
FILE SECTION.
FD  CUSTOMERSDATA.
COPY projectCusDataBook.

FD ACCOUNT.
01 ACCOUNT-NUMBERS            PIC 9(10).

WORKING-STORAGE SECTION.
COPY pojCusRecordBook.
01 FILE-STATUS                PIC XX.
01 END-OF-FILE                PIC X VALUE "N".
01 TEMP-ACCOUNT-NUMBERS       PIC 9(10).
01 ALIGN                      PIC X(2) VALUE " ".
01  WS-CUSTOMER-COUNTER   PIC 9(2) VALUE 1.
01  WS-CUS-NAMES          PIC X(50) VALUE SPACES.
01  WS-ACCOUNT-TYPES      PIC X(6) VALUE SPACES.
01  Random-Number    PIC 9(1)V9(9) VALUE 0.
01  Scaled-Random-Number PIC 9(3) VALUE 0.
```



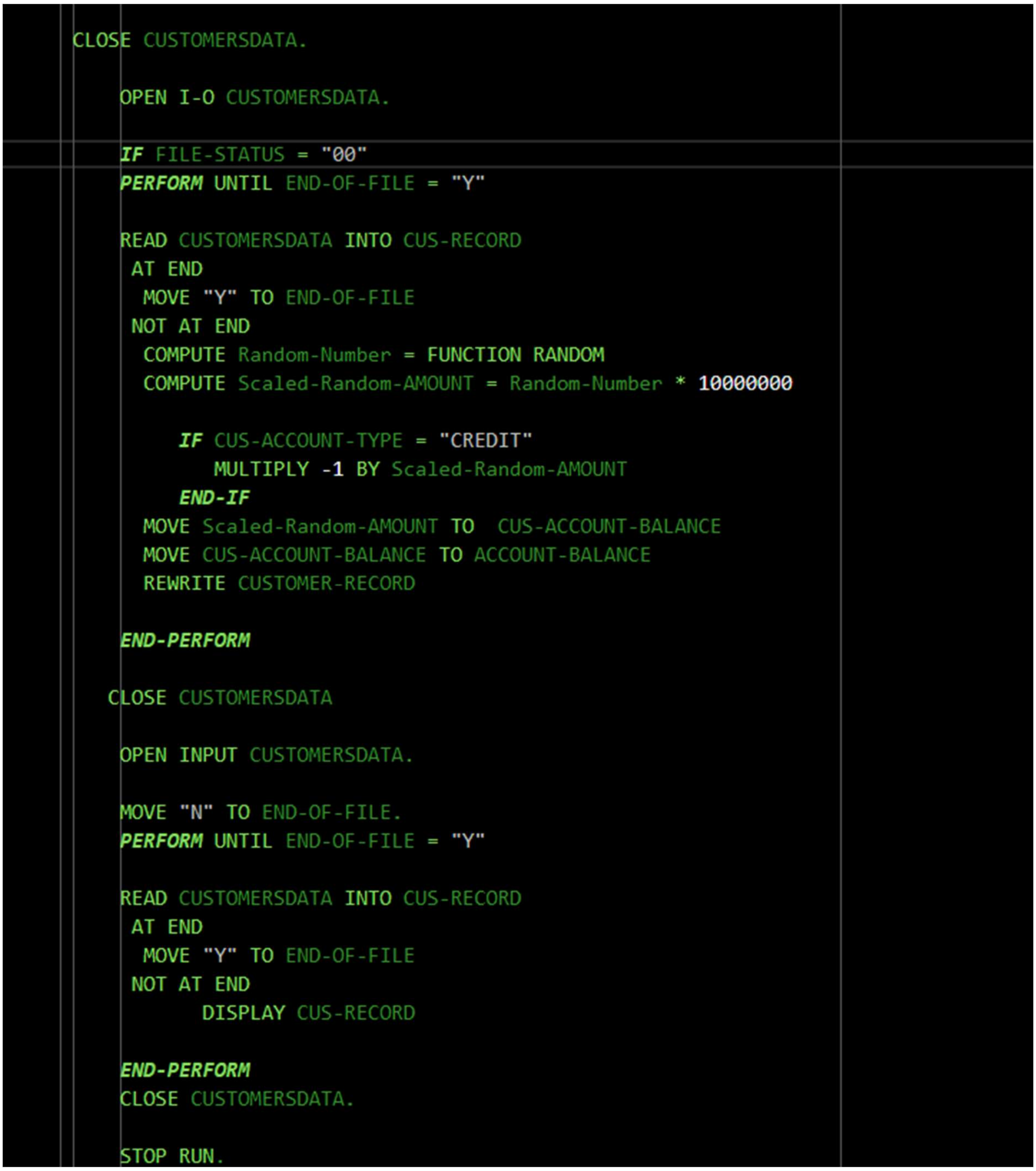
```
DATA DIVISION.
WORKING-STORAGE SECTION.
01  WS-CUSTOMER-COUNTER   PIC 9(2) VALUE 1.
01  WS-CUS-NAMES          PIC X(50) VALUE SPACES.
01  WS-ACCOUNT-TYPES      PIC X(6) VALUE SPACES.
01  Random-Number    PIC 9(1)V9(9) VALUE 0.
01  Scaled-Random-Number PIC 9(3) VALUE 0.
01  Scaled-Random-AMOUNT PIC S9(9)V9(2).
```



```
PROGRAM-ID. CREATE-CUSTOMERSDATA.
PROCEDURE DIVISION.

OPEN INPUT CUSTOMERSDATA.
IF FILE-STATUS = "00"
 DISPLAY "CUSTOMERSDATA exists and opened successfully."

ELSE IF FILE-STATUS = "35"

 OPEN OUTPUT CUSTOMERSDATA
 OPEN INPUT ACCOUNT
 PERFORM UNTIL END-OF-FILE = 'Y'

    READ ACCOUNT INTO TEMP-ACCOUNT-NUMBERS

      AT END
        MOVE 'Y' TO END-OF-FILE

      NOT AT END
        MOVE TEMP-ACCOUNT-NUMBERS TO ACCOUNT-NUM
    END-READ

  EVALUATE WS-CUSTOMER-COUNTER
  WHEN 1 MOVE 'John Doe                         ' TO WS-CUS-NAMES
  WHEN 2 MOVE 'Jane Smith                       ' TO WS-CUS-NAMES
  WHEN 3 MOVE 'Alice Johnson                    ' TO WS-CUS-NAMES
  WHEN 4 MOVE 'Bob Brown                        ' TO WS-CUS-NAMES
  WHEN 5 MOVE 'Charlie Davis                    ' TO WS-CUS-NAMES
  WHEN 6 MOVE 'Diana Clark                      ' TO WS-CUS-NAMES
  WHEN 7 MOVE 'Ethan Lewis                      ' TO WS-CUS-NAMES
  WHEN 8 MOVE 'Fiona Adams                      ' TO WS-CUS-NAMES
  WHEN 9 MOVE 'George Young                     ' TO WS-CUS-NAMES
  WHEN 10 MOVE 'Hannah White                    ' TO WS-CUS-NAMES
  WHEN 11 MOVE 'Ian Scott                       ' TO WS-CUS-NAMES
  WHEN 12 MOVE 'Julia Green                     ' TO WS-CUS-NAMES
  WHEN 13 MOVE 'Kevin Hill                      ' TO WS-CUS-NAMES
  WHEN 14 MOVE 'Lily Cooper                     ' TO WS-CUS-NAMES
  WHEN 15 MOVE 'Mason Carter                    ' TO WS-CUS-NAMES
  WHEN 16 MOVE 'Nora Evans                      ' TO WS-CUS-NAMES
  WHEN 17 MOVE 'Oliver Foster                   ' TO WS-CUS-NAMES
  WHEN 18 MOVE 'Paula Baker                     ' TO WS-CUS-NAMES
  WHEN 19 MOVE 'Quincy Turner                   ' TO WS-CUS-NAMES
  WHEN 20 MOVE 'Ruby Hall                       ' TO WS-CUS-NAMES
```

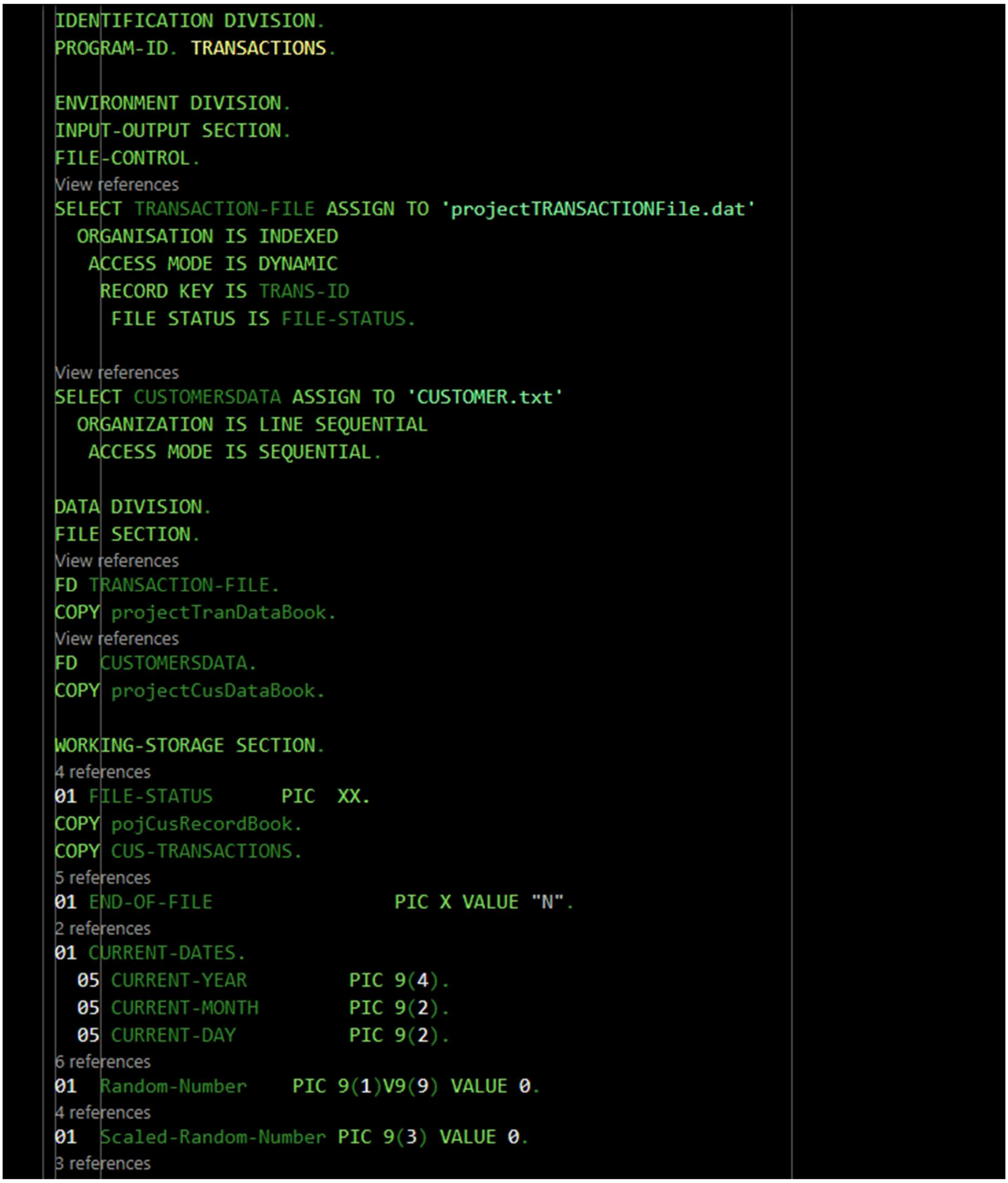


```
 WHEN 20 MOVE 'Ruby Hall                        ' TO WS-CUS-NAMES
WHEN OTHER MOVE 'Unknown                        ' TO WS-CUS-NAMES

END-EVALUATE
MOVE WS-CUS-NAMES TO ACCOUNT-NAMES
COMPUTE ACCOUNT-BALANCE = 0

COMPUTE Random-Number = FUNCTION RANDOM
COMPUTE Scaled-Random-Number = Random-Number * 2

EVALUATE Scaled-Random-Number
   WHEN 0 MOVE 'DEBIT ' TO ACCOUNT-TYPE
   WHEN 1 MOVE 'CREDIT' TO ACCOUNT-TYPE
END-EVALUATE

 WRITE CUSTOMER-RECORD
ADD 1 TO WS-CUSTOMER-COUNTER

  IF WS-CUSTOMER-COUNTER > 20
       MOVE 'Y' TO END-OF-FILE
  END-IF

 END-PERFORM

  CLOSE ACCOUNT
  CLOSE CUSTOMERSDATA

ELSE
  DISPLAY "Error opening file. FILE STATUS: " FILE-STATUS

 END-IF.
```

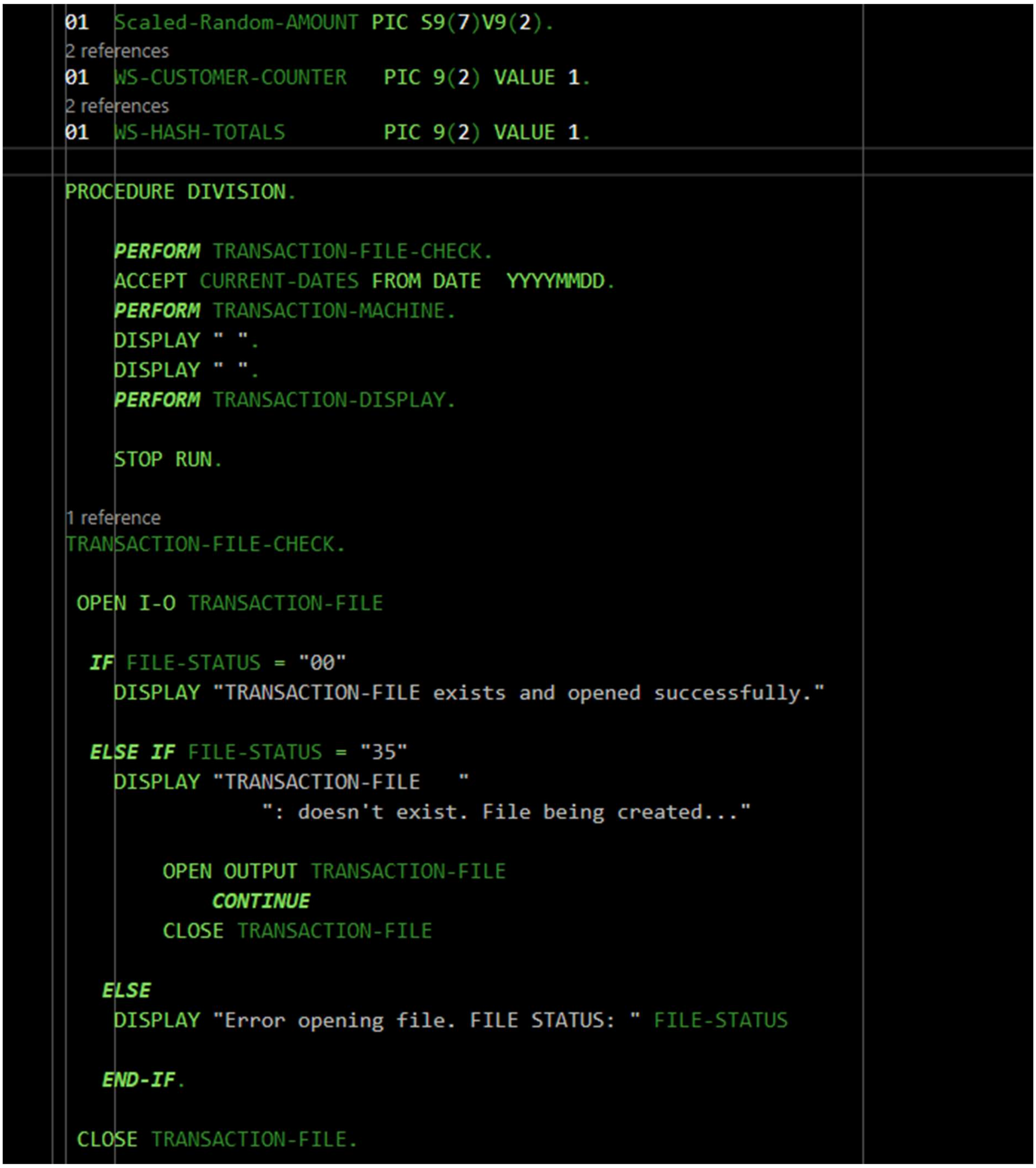


```
CLOSE CUSTOMERSDATA.

    OPEN I-O CUSTOMERSDATA.

    IF FILE-STATUS = "00"
    PERFORM UNTIL END-OF-FILE = "Y"

    READ CUSTOMERSDATA INTO CUS-RECORD
     AT END
      MOVE "Y" TO END-OF-FILE
     NOT AT END
      COMPUTE Random-Number = FUNCTION RANDOM
      COMPUTE Scaled-Random-AMOUNT = Random-Number * 10000000

         IF CUS-ACCOUNT-TYPE = "CREDIT"
            MULTIPLY -1 BY Scaled-Random-AMOUNT
         END-IF
      MOVE Scaled-Random-AMOUNT TO  CUS-ACCOUNT-BALANCE
      MOVE CUS-ACCOUNT-BALANCE TO ACCOUNT-BALANCE
      REWRITE CUSTOMER-RECORD

    END-PERFORM

   CLOSE CUSTOMERSDATA

    OPEN INPUT CUSTOMERSDATA.

    MOVE "N" TO END-OF-FILE.
    PERFORM UNTIL END-OF-FILE = "Y"

    READ CUSTOMERSDATA INTO CUS-RECORD
     AT END
      MOVE "Y" TO END-OF-FILE
     NOT AT END
           DISPLAY CUS-RECORD

    END-PERFORM
    CLOSE CUSTOMERSDATA.

    STOP RUN.
```

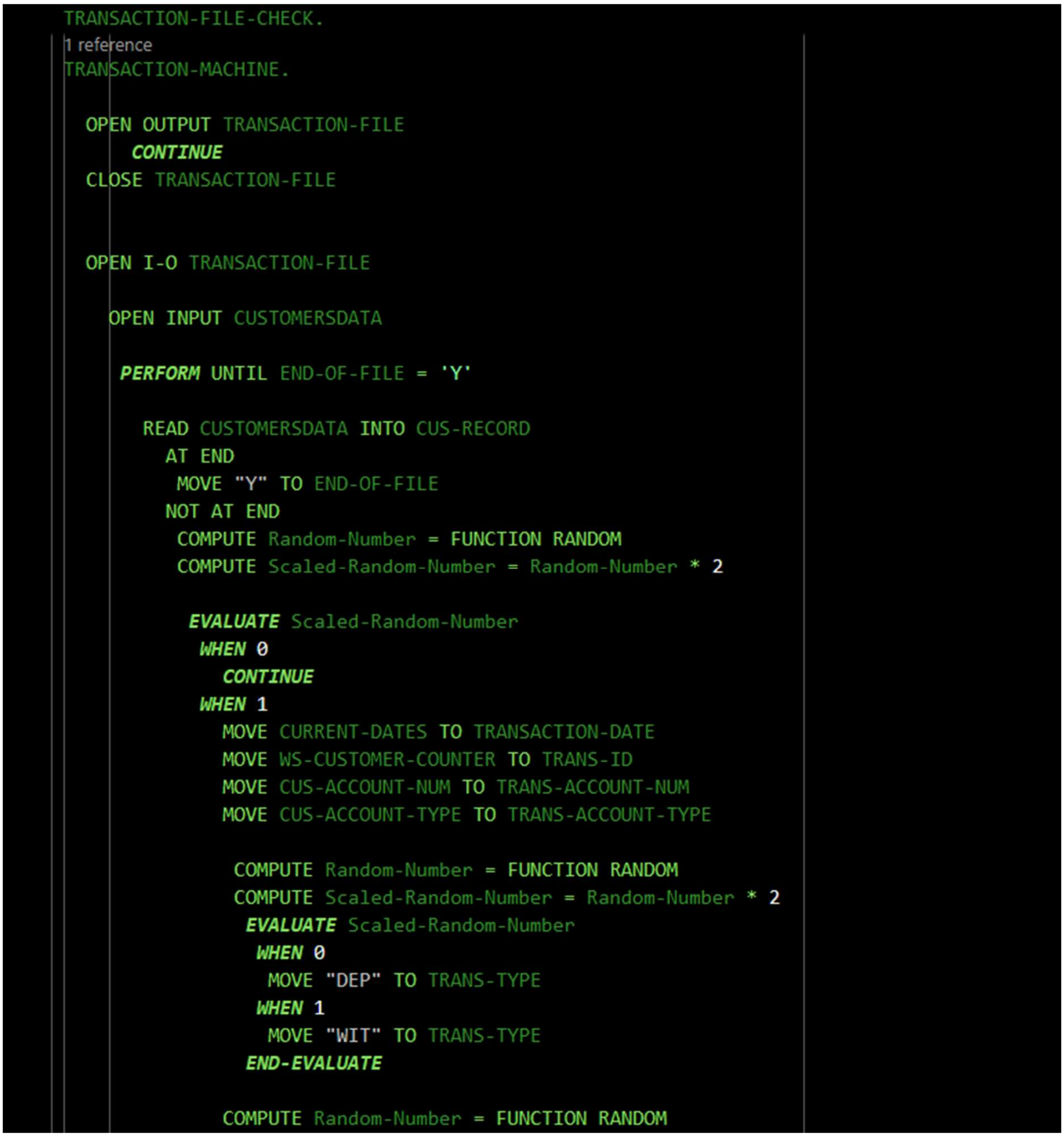
1.TRANSACTION generation



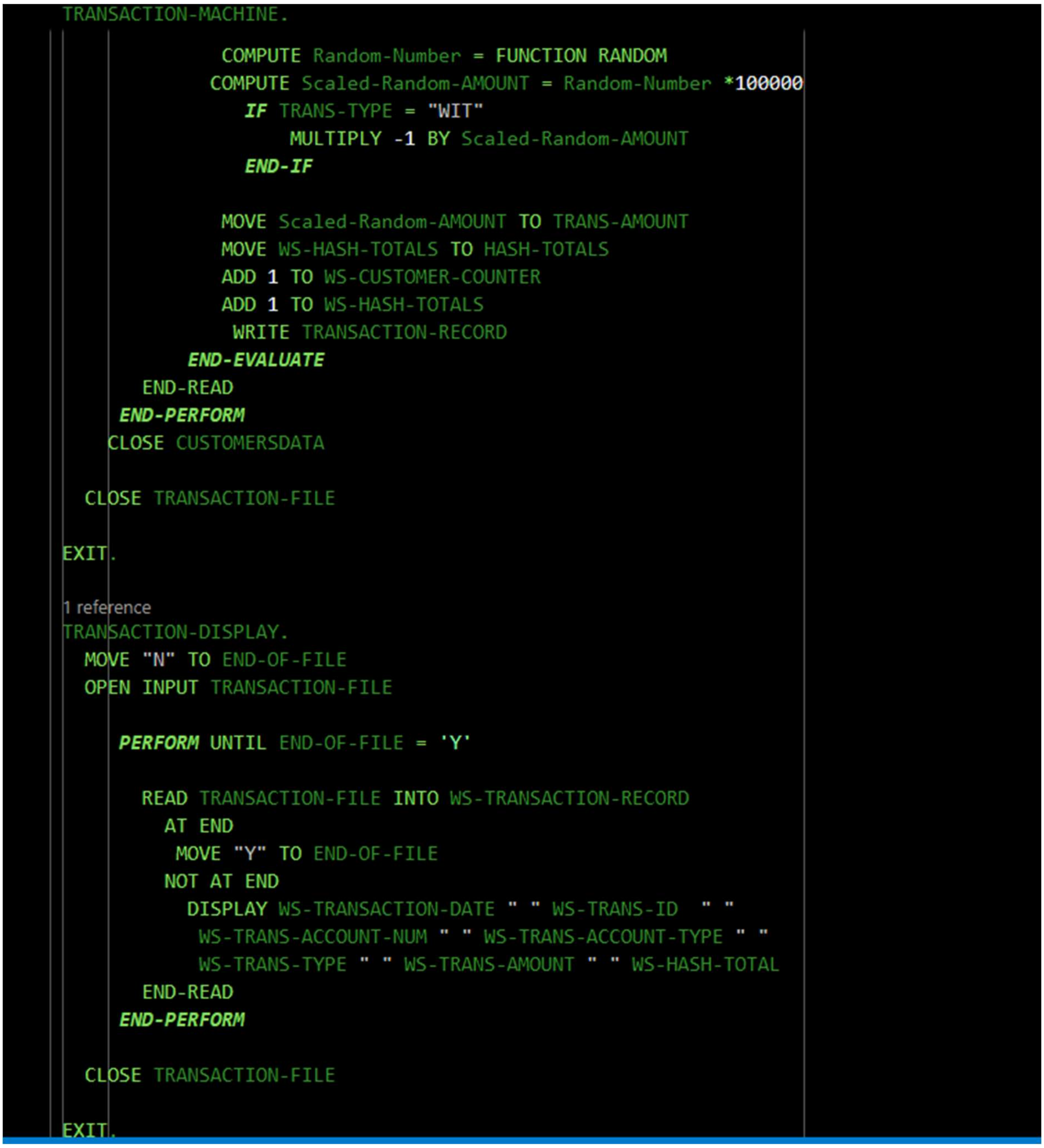
```
IDENTIFICATION DIVISION.
PROGRAM-ID. TRANSACTIONS.

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
View references
SELECT TRANSACTION-FILE ASSIGN TO 'projectTRANSACTIONFile.dat'
  ORGANISATION IS INDEXED
   ACCESS MODE IS DYNAMIC
    RECORD KEY IS TRANS-ID
     FILE STATUS IS FILE-STATUS.

View references
SELECT CUSTOMERSDATA ASSIGN TO 'CUSTOMER.txt'
  ORGANIZATION IS LINE SEQUENTIAL
   ACCESS MODE IS SEQUENTIAL.

DATA DIVISION.
FILE SECTION.
View references
FD TRANSACTION-FILE.
COPY projectTranDataBook.
View references
FD  CUSTOMERSDATA.
COPY projectCusDataBook.

WORKING-STORAGE SECTION.
4 references
01 FILE-STATUS      PIC  XX.
COPY pojCusRecordBook.
COPY CUS-TRANSACTIONS.
5 references
01 END-OF-FILE                PIC X VALUE "N".
2 references
01 CURRENT-DATES.
  05 CURRENT-YEAR         PIC 9(4).
  05 CURRENT-MONTH        PIC 9(2).
  05 CURRENT-DAY          PIC 9(2).
6 references
01  Random-Number    PIC 9(1)V9(9) VALUE 0.
4 references
01  Scaled-Random-Number PIC 9(3) VALUE 0.
3 references
```



```
01  Scaled-Random-AMOUNT PIC S9(7)V9(2).
2 references
01  WS-CUSTOMER-COUNTER   PIC 9(2) VALUE 1.
2 references
01  WS-HASH-TOTALS        PIC 9(2) VALUE 1.

PROCEDURE DIVISION.

    PERFORM TRANSACTION-FILE-CHECK.
    ACCEPT CURRENT-DATES FROM DATE  YYYYMMDD.
    PERFORM TRANSACTION-MACHINE.
    DISPLAY " ".
    DISPLAY " ".
    PERFORM TRANSACTION-DISPLAY.

    STOP RUN.

1 reference
TRANSACTION-FILE-CHECK.

 OPEN I-O TRANSACTION-FILE

  IF FILE-STATUS = "00"
    DISPLAY "TRANSACTION-FILE exists and opened successfully."

  ELSE IF FILE-STATUS = "35"
    DISPLAY "TRANSACTION-FILE   "
                ": doesn't exist. File being created..."

        OPEN OUTPUT TRANSACTION-FILE
            CONTINUE
        CLOSE TRANSACTION-FILE

   ELSE
    DISPLAY "Error opening file. FILE STATUS: " FILE-STATUS

   END-IF.

 CLOSE TRANSACTION-FILE.
```

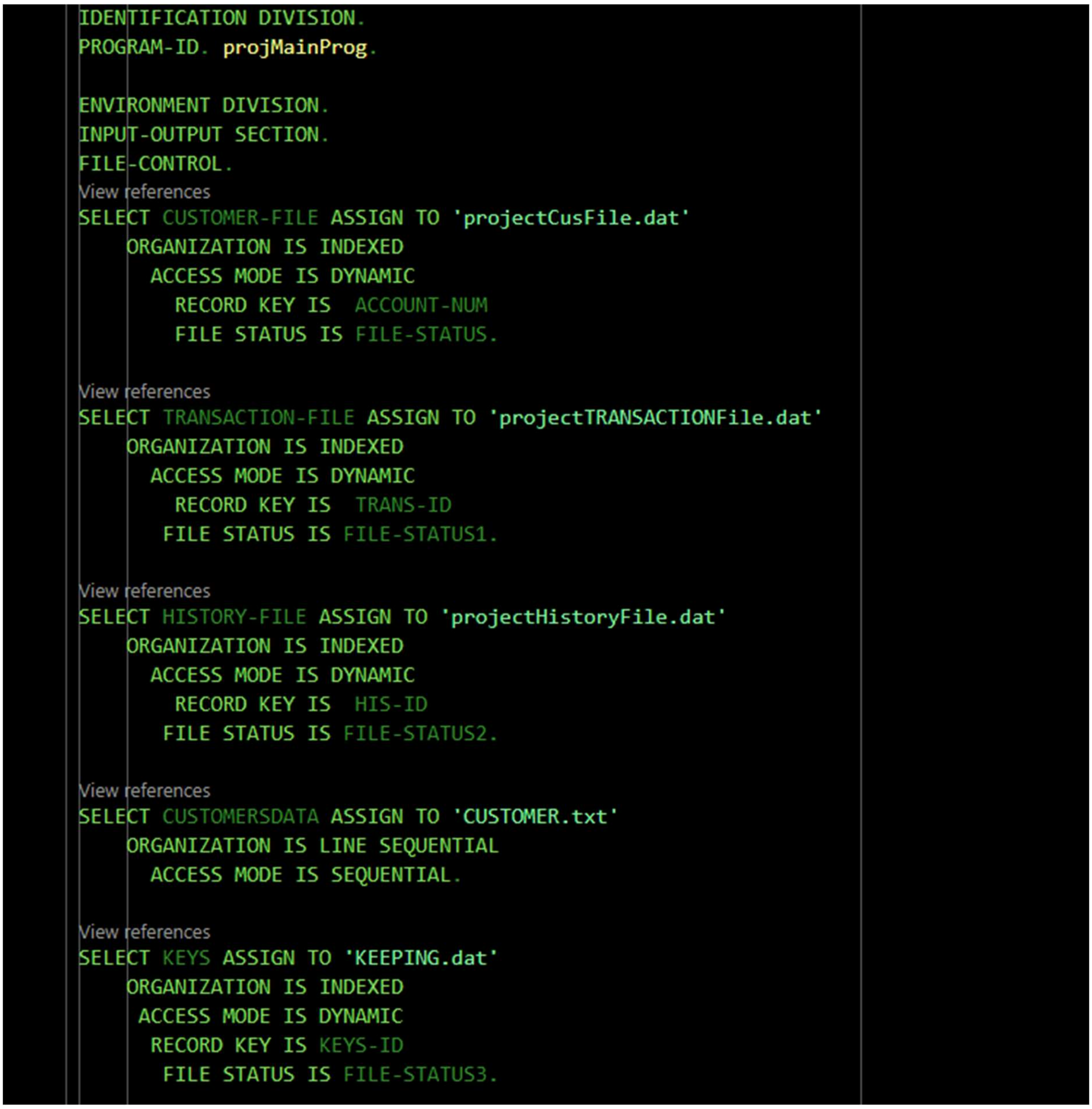


```
TRANSACTION-FILE-CHECK.
1 reference
TRANSACTION-MACHINE.

  OPEN OUTPUT TRANSACTION-FILE
      CONTINUE
  CLOSE TRANSACTION-FILE


  OPEN I-O TRANSACTION-FILE

    OPEN INPUT CUSTOMERSDATA

     PERFORM UNTIL END-OF-FILE = 'Y'

       READ CUSTOMERSDATA INTO CUS-RECORD
         AT END
          MOVE "Y" TO END-OF-FILE
         NOT AT END
          COMPUTE Random-Number = FUNCTION RANDOM
          COMPUTE Scaled-Random-Number = Random-Number * 2

           EVALUATE Scaled-Random-Number
            WHEN 0
              CONTINUE
            WHEN 1
              MOVE CURRENT-DATES TO TRANSACTION-DATE
              MOVE WS-CUSTOMER-COUNTER TO TRANS-ID
              MOVE CUS-ACCOUNT-NUM TO TRANS-ACCOUNT-NUM
              MOVE CUS-ACCOUNT-TYPE TO TRANS-ACCOUNT-TYPE

               COMPUTE Random-Number = FUNCTION RANDOM
               COMPUTE Scaled-Random-Number = Random-Number * 2
                EVALUATE Scaled-Random-Number
                 WHEN 0
                  MOVE "DEP" TO TRANS-TYPE
                 WHEN 1
                  MOVE "WIT" TO TRANS-TYPE
                END-EVALUATE

              COMPUTE Random-Number = FUNCTION RANDOM
```

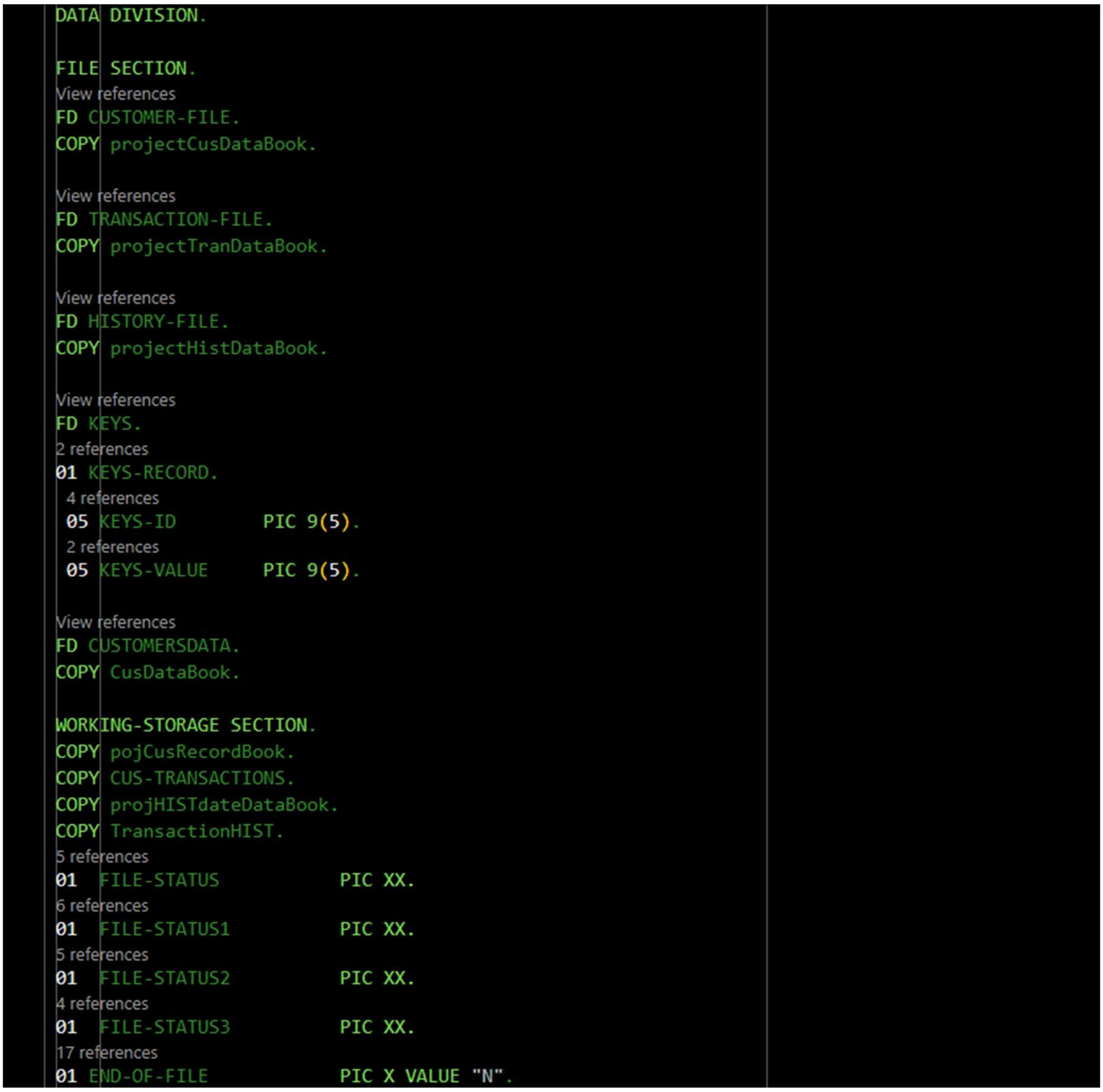


```
TRANSACTION-MACHINE.
              COMPUTE Random-Number = FUNCTION RANDOM
             COMPUTE Scaled-Random-AMOUNT = Random-Number *100000
                IF TRANS-TYPE = "WIT"
                    MULTIPLY -1 BY Scaled-Random-AMOUNT
                END-IF

              MOVE Scaled-Random-AMOUNT TO TRANS-AMOUNT
              MOVE WS-HASH-TOTALS TO HASH-TOTALS
              ADD 1 TO WS-CUSTOMER-COUNTER
              ADD 1 TO WS-HASH-TOTALS
               WRITE TRANSACTION-RECORD
           END-EVALUATE
       END-READ
     END-PERFORM
    CLOSE CUSTOMERSDATA

  CLOSE TRANSACTION-FILE

EXIT.

1 reference
TRANSACTION-DISPLAY.
  MOVE "N" TO END-OF-FILE
  OPEN INPUT TRANSACTION-FILE

     PERFORM UNTIL END-OF-FILE = 'Y'

       READ TRANSACTION-FILE INTO WS-TRANSACTION-RECORD
         AT END
          MOVE "Y" TO END-OF-FILE
         NOT AT END
           DISPLAY WS-TRANSACTION-DATE " " WS-TRANS-ID  " "
            WS-TRANS-ACCOUNT-NUM " " WS-TRANS-ACCOUNT-TYPE " "
            WS-TRANS-TYPE " " WS-TRANS-AMOUNT " " WS-HASH-TOTAL
       END-READ
     END-PERFORM

  CLOSE TRANSACTION-FILE

EXIT.
```

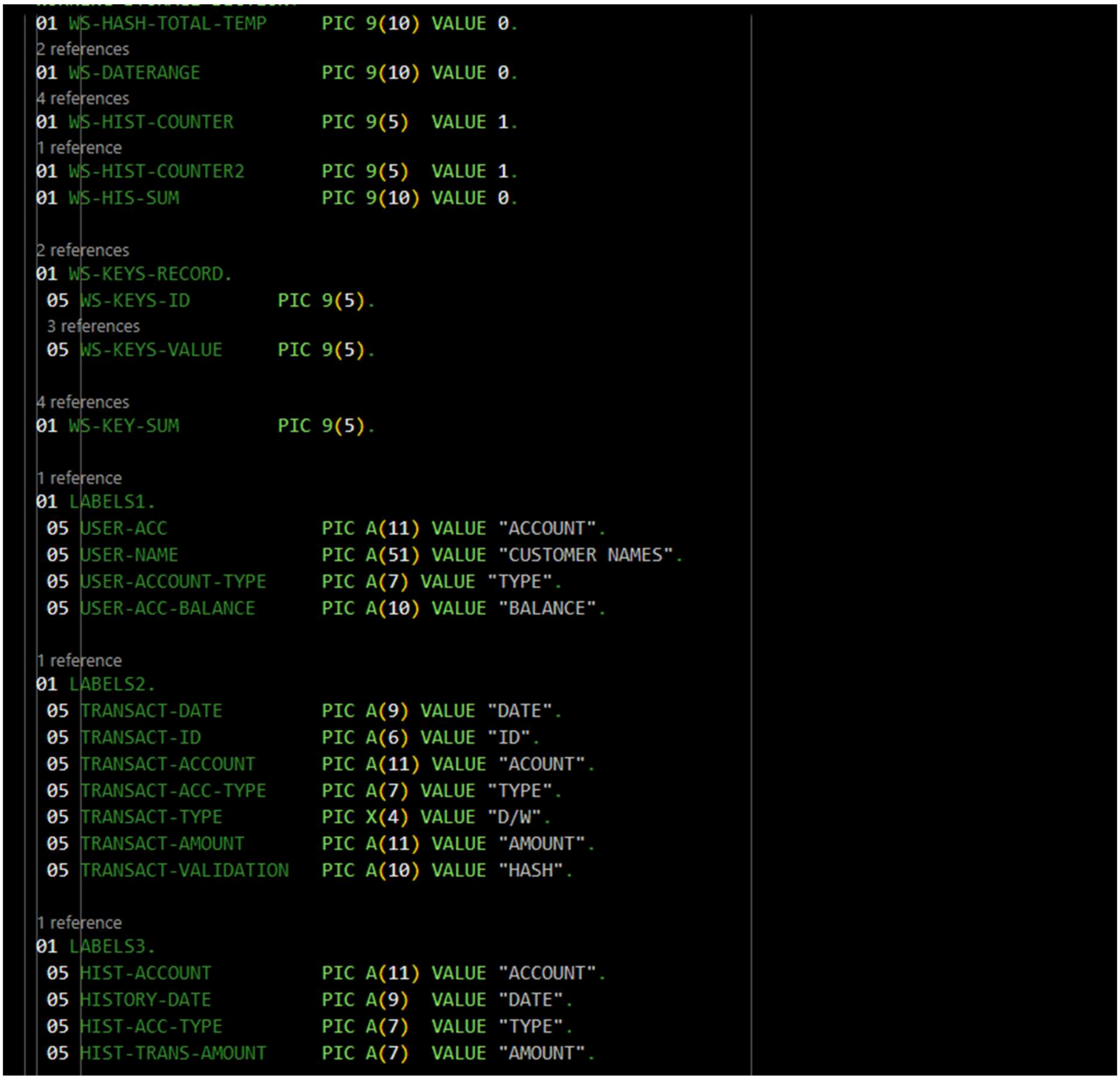
1.MAIN CODE



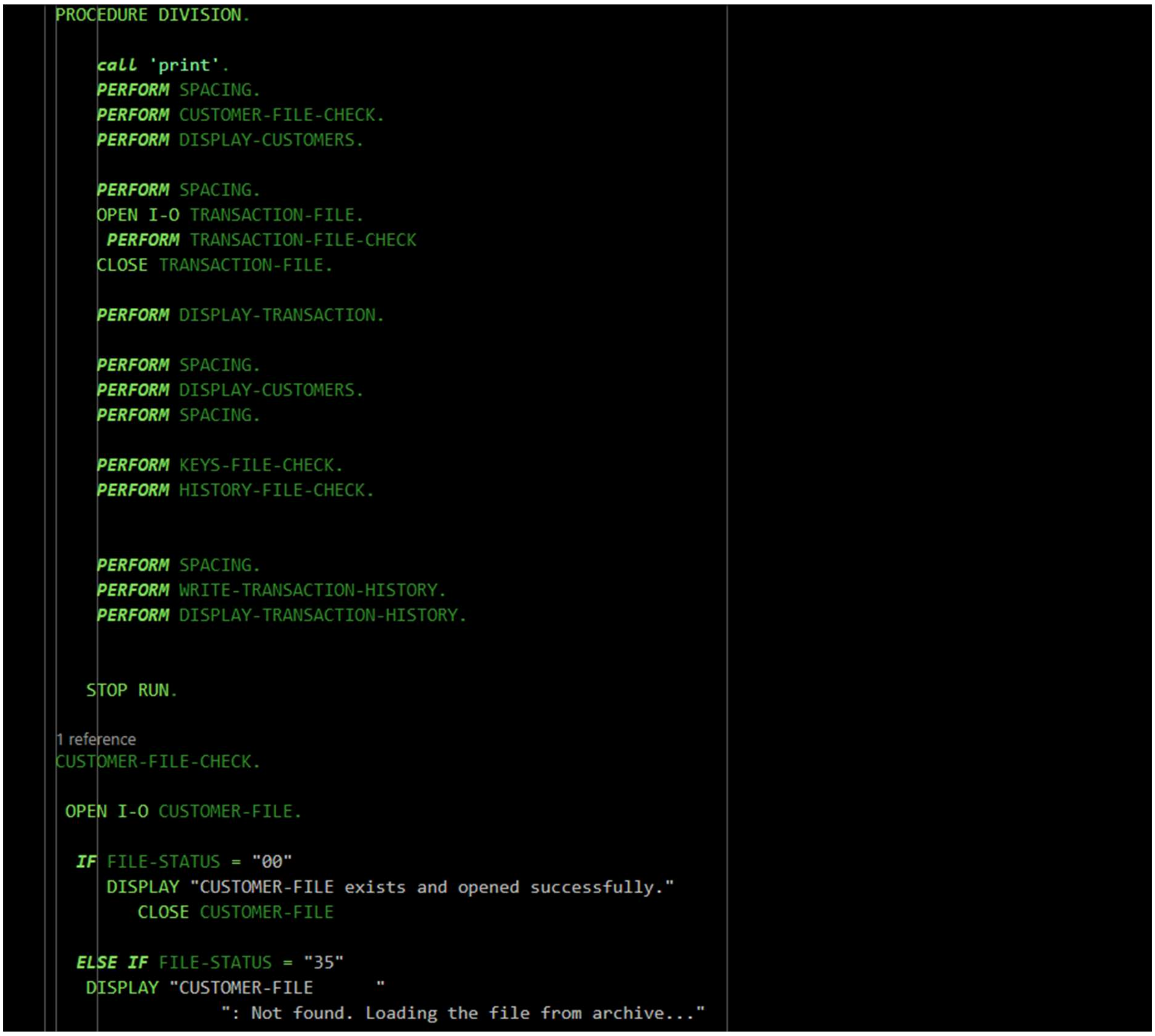
```
IDENTIFICATION DIVISION.
PROGRAM-ID. projMainProg.

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
View references
SELECT CUSTOMER-FILE ASSIGN TO 'projectCusFile.dat'
    ORGANIZATION IS INDEXED
      ACCESS MODE IS DYNAMIC
        RECORD KEY IS  ACCOUNT-NUM
        FILE STATUS IS FILE-STATUS.

View references
SELECT TRANSACTION-FILE ASSIGN TO 'projectTRANSACTIONFile.dat'
    ORGANIZATION IS INDEXED
      ACCESS MODE IS DYNAMIC
        RECORD KEY IS  TRANS-ID
       FILE STATUS IS FILE-STATUS1.

View references
SELECT HISTORY-FILE ASSIGN TO 'projectHistoryFile.dat'
    ORGANIZATION IS INDEXED
      ACCESS MODE IS DYNAMIC
        RECORD KEY IS  HIS-ID
       FILE STATUS IS FILE-STATUS2.

View references
SELECT CUSTOMERSDATA ASSIGN TO 'CUSTOMER.txt'
    ORGANIZATION IS LINE SEQUENTIAL
      ACCESS MODE IS SEQUENTIAL.

View references
SELECT KEYS ASSIGN TO 'KEEPING.dat'
    ORGANIZATION IS INDEXED
     ACCESS MODE IS DYNAMIC
      RECORD KEY IS KEYS-ID
       FILE STATUS IS FILE-STATUS3.
```



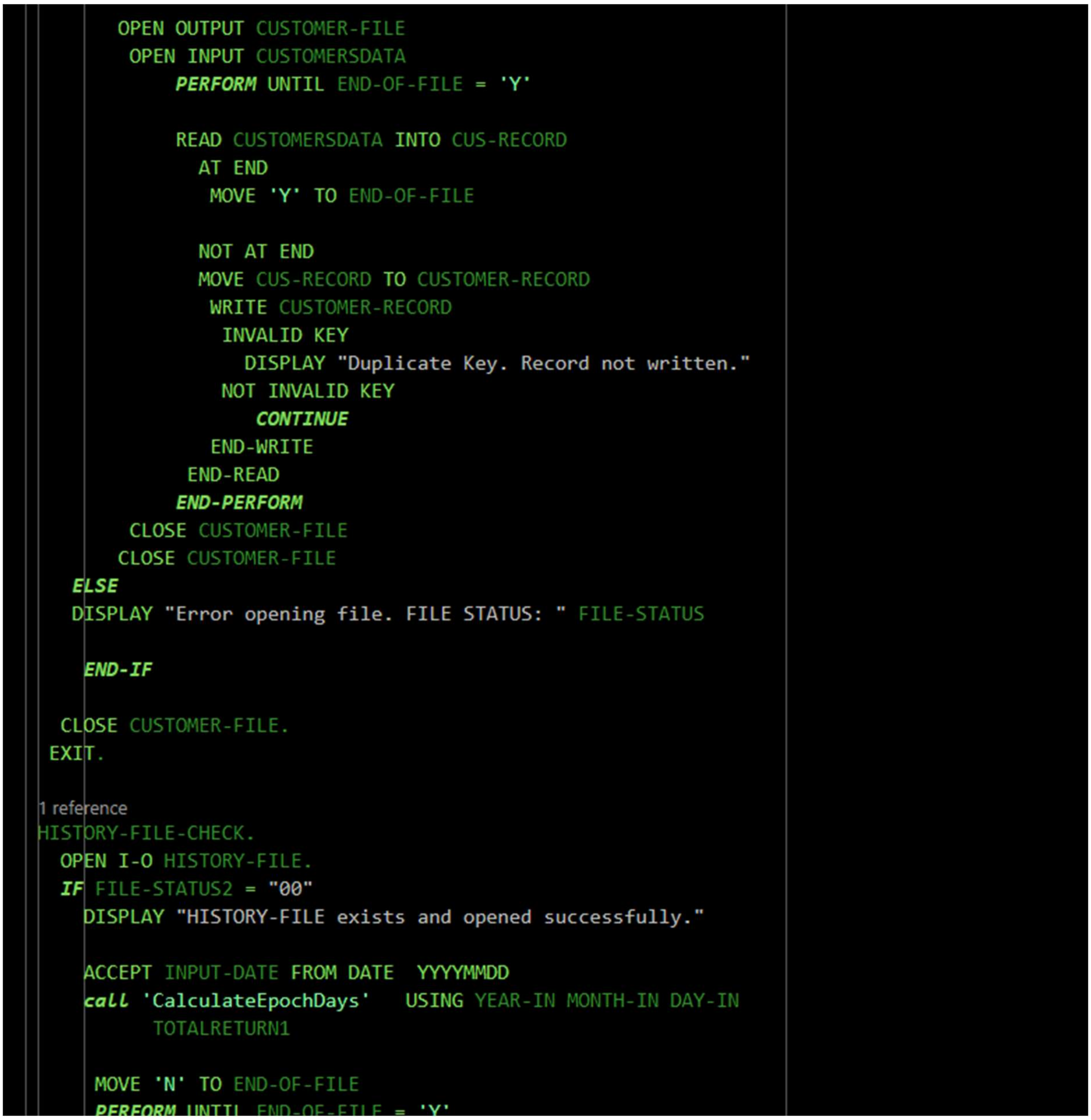
```
DATA DIVISION.

FILE SECTION.
View references
FD CUSTOMER-FILE.
COPY projectCusDataBook.

View references
FD TRANSACTION-FILE.
COPY projectTranDataBook.

View references
FD HISTORY-FILE.
COPY projectHistDataBook.

View references
FD KEYS.
2 references
01 KEYS-RECORD.
 4 references
 05 KEYS-ID        PIC 9(5).
 2 references
 05 KEYS-VALUE     PIC 9(5).

View references
FD CUSTOMERSDATA.
COPY CusDataBook.

WORKING-STORAGE SECTION.
COPY pojCusRecordBook.
COPY CUS-TRANSACTIONS.
COPY projHISTdateDataBook.
COPY TransactionHIST.
5 references
01   FILE-STATUS             PIC XX.
6 references
01   FILE-STATUS1            PIC XX.
5 references
01   FILE-STATUS2            PIC XX.
4 references
01   FILE-STATUS3            PIC XX.
17 references
01 END-OF-FILE               PIC X VALUE "N".
```



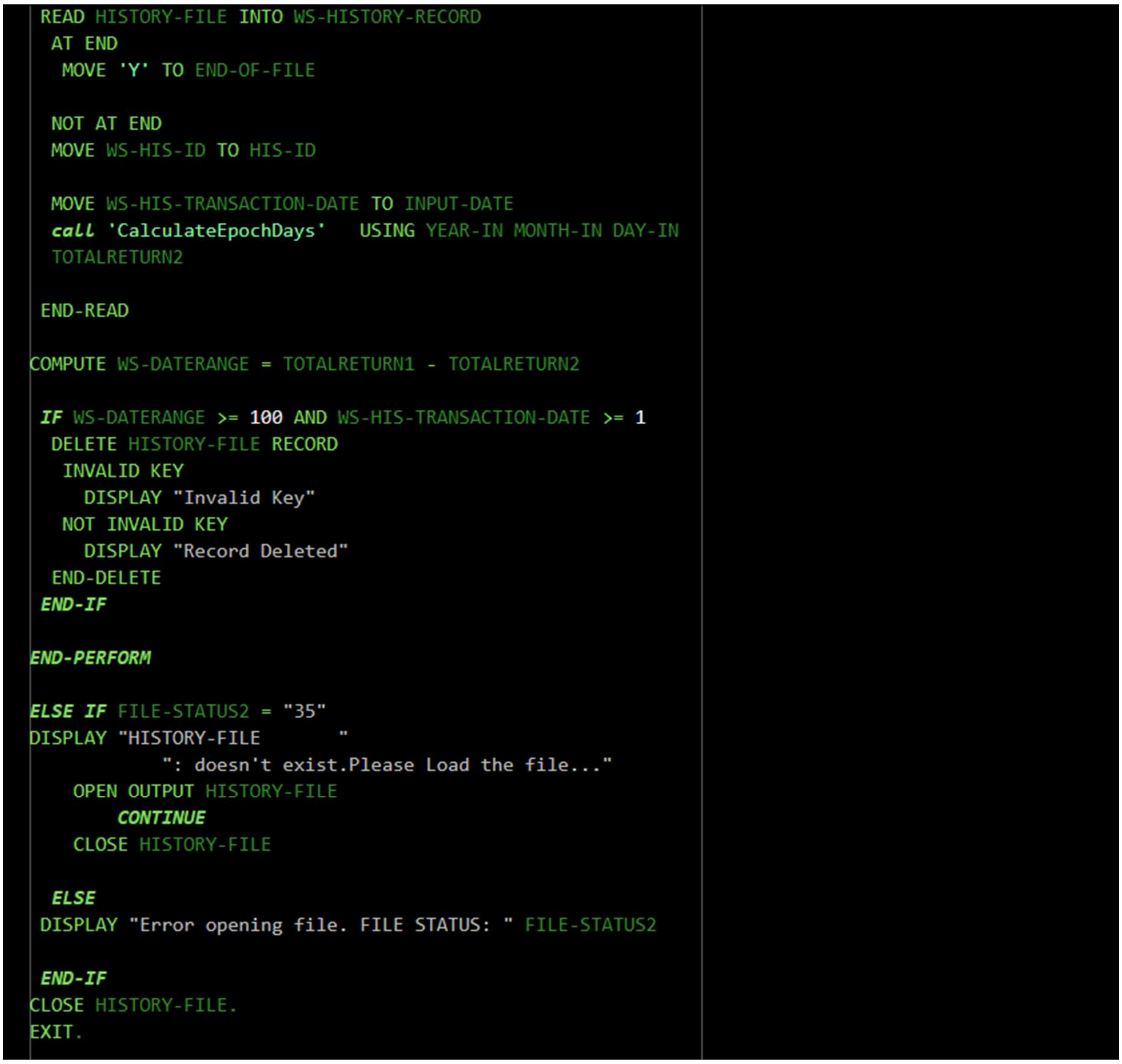
```
01 WS-HASH-TOTAL-TEMP      PIC 9(10) VALUE 0.
2 references
01 WS-DATERANGE            PIC 9(10) VALUE 0.
4 references
01 WS-HIST-COUNTER         PIC 9(5)  VALUE 1.
1 reference
01 WS-HIST-COUNTER2        PIC 9(5)  VALUE 1.
01 WS-HIS-SUM              PIC 9(10) VALUE 0.

2 references
01 WS-KEYS-RECORD.
 05 WS-KEYS-ID         PIC 9(5).
 3 references
 05 WS-KEYS-VALUE      PIC 9(5).

4 references
01 WS-KEY-SUM          PIC 9(5).

1 reference
01 LABELS1.
 05 USER-ACC               PIC A(11) VALUE "ACCOUNT".
 05 USER-NAME              PIC A(51) VALUE "CUSTOMER NAMES".
 05 USER-ACCOUNT-TYPE      PIC A(7) VALUE "TYPE".
 05 USER-ACC-BALANCE       PIC A(10) VALUE "BALANCE".

1 reference
01 LABELS2.
 05 TRANSACT-DATE          PIC A(9) VALUE "DATE".
 05 TRANSACT-ID            PIC A(6) VALUE "ID".
 05 TRANSACT-ACCOUNT       PIC A(11) VALUE "ACOUNT".
 05 TRANSACT-ACC-TYPE      PIC A(7) VALUE "TYPE".
 05 TRANSACT-TYPE          PIC X(4) VALUE "D/W".
 05 TRANSACT-AMOUNT        PIC A(11) VALUE "AMOUNT".
 05 TRANSACT-VALIDATION    PIC A(10) VALUE "HASH".

1 reference
01 LABELS3.
 05 HIST-ACCOUNT           PIC A(11) VALUE "ACCOUNT".
 05 HISTORY-DATE           PIC A(9)  VALUE "DATE".
 05 HIST-ACC-TYPE          PIC A(7)  VALUE "TYPE".
 05 HIST-TRANS-AMOUNT      PIC A(7)  VALUE "AMOUNT".
```



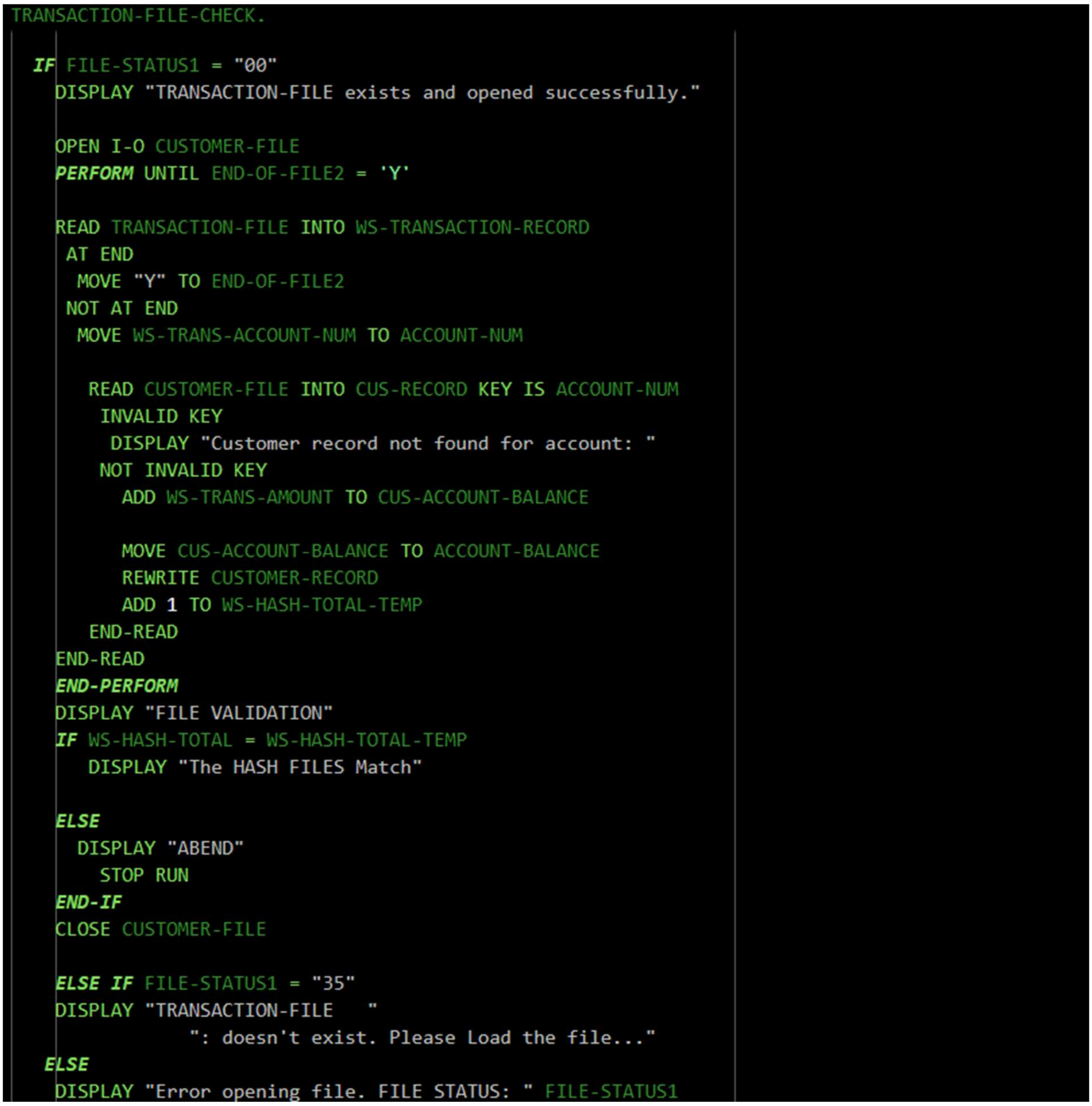
```
PROCEDURE DIVISION.

    call 'print'.
    PERFORM SPACING.
    PERFORM CUSTOMER-FILE-CHECK.
    PERFORM DISPLAY-CUSTOMERS.

    PERFORM SPACING.
    OPEN I-O TRANSACTION-FILE.
     PERFORM TRANSACTION-FILE-CHECK
    CLOSE TRANSACTION-FILE.

    PERFORM DISPLAY-TRANSACTION.

    PERFORM SPACING.
    PERFORM DISPLAY-CUSTOMERS.
    PERFORM SPACING.

    PERFORM KEYS-FILE-CHECK.
    PERFORM HISTORY-FILE-CHECK.


    PERFORM SPACING.
    PERFORM WRITE-TRANSACTION-HISTORY.
    PERFORM DISPLAY-TRANSACTION-HISTORY.


   STOP RUN.

1 reference
CUSTOMER-FILE-CHECK.

 OPEN I-O CUSTOMER-FILE.

  IF FILE-STATUS = "00"
     DISPLAY "CUSTOMER-FILE exists and opened successfully."
        CLOSE CUSTOMER-FILE

  ELSE IF FILE-STATUS = "35"
   DISPLAY "CUSTOMER-FILE      "
                ": Not found. Loading the file from archive..."
```



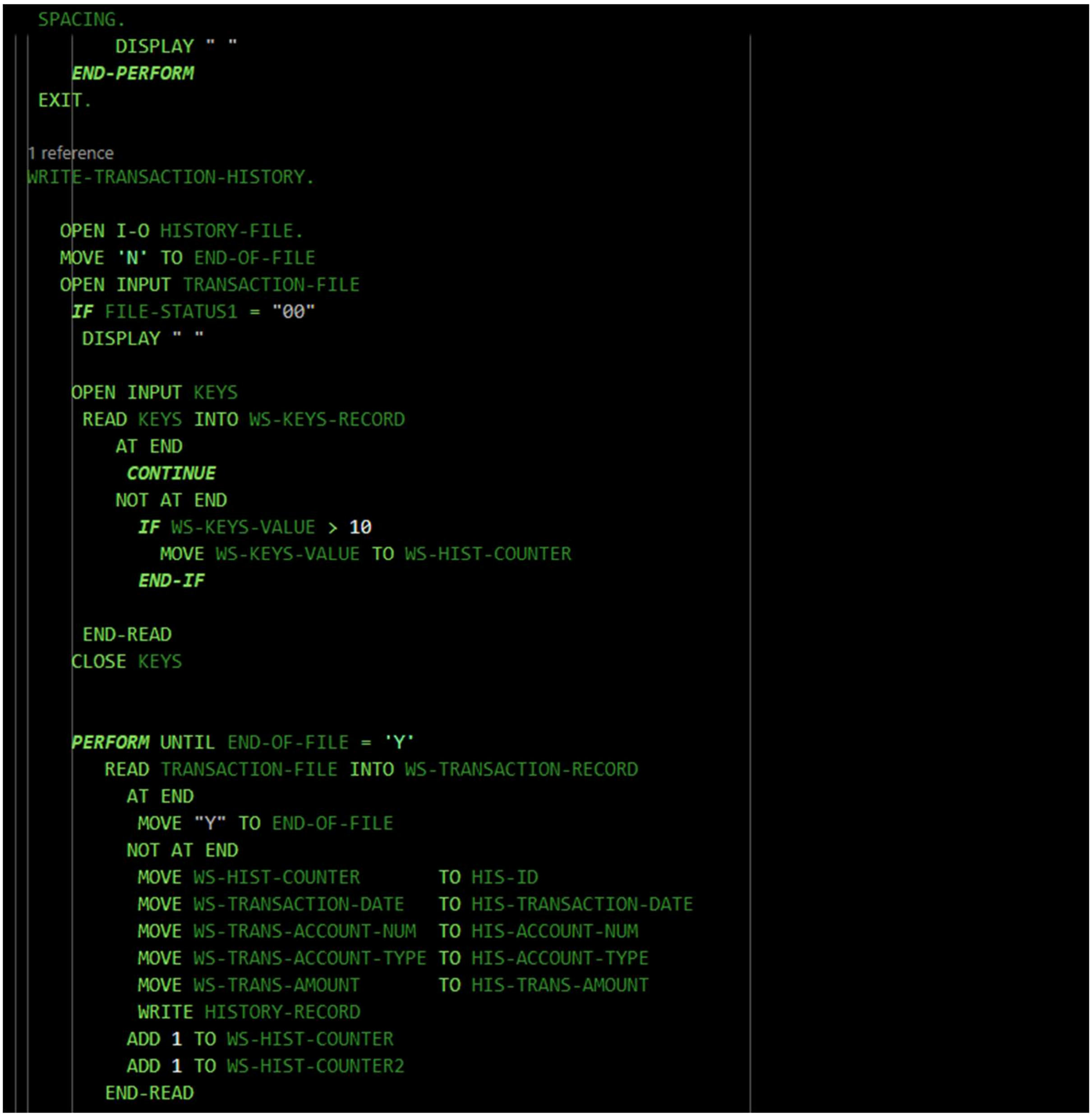
```
       OPEN OUTPUT CUSTOMER-FILE
        OPEN INPUT CUSTOMERSDATA
            PERFORM UNTIL END-OF-FILE = 'Y'

            READ CUSTOMERSDATA INTO CUS-RECORD
              AT END
               MOVE 'Y' TO END-OF-FILE

              NOT AT END
              MOVE CUS-RECORD TO CUSTOMER-RECORD
               WRITE CUSTOMER-RECORD
                INVALID KEY
                  DISPLAY "Duplicate Key. Record not written."
                NOT INVALID KEY
                   CONTINUE
               END-WRITE
             END-READ
            END-PERFORM
        CLOSE CUSTOMER-FILE
       CLOSE CUSTOMER-FILE
   ELSE
   DISPLAY "Error opening file. FILE STATUS: " FILE-STATUS

    END-IF

  CLOSE CUSTOMER-FILE.
 EXIT.

1 reference
HISTORY-FILE-CHECK.
  OPEN I-O HISTORY-FILE.
  IF FILE-STATUS2 = "00"
    DISPLAY "HISTORY-FILE exists and opened successfully."

    ACCEPT INPUT-DATE FROM DATE  YYYYMMDD
    call 'CalculateEpochDays'   USING YEAR-IN MONTH-IN DAY-IN
          TOTALRETURN1

     MOVE 'N' TO END-OF-FILE
     PERFORM UNTIL END-OF-FILE = 'Y'
```



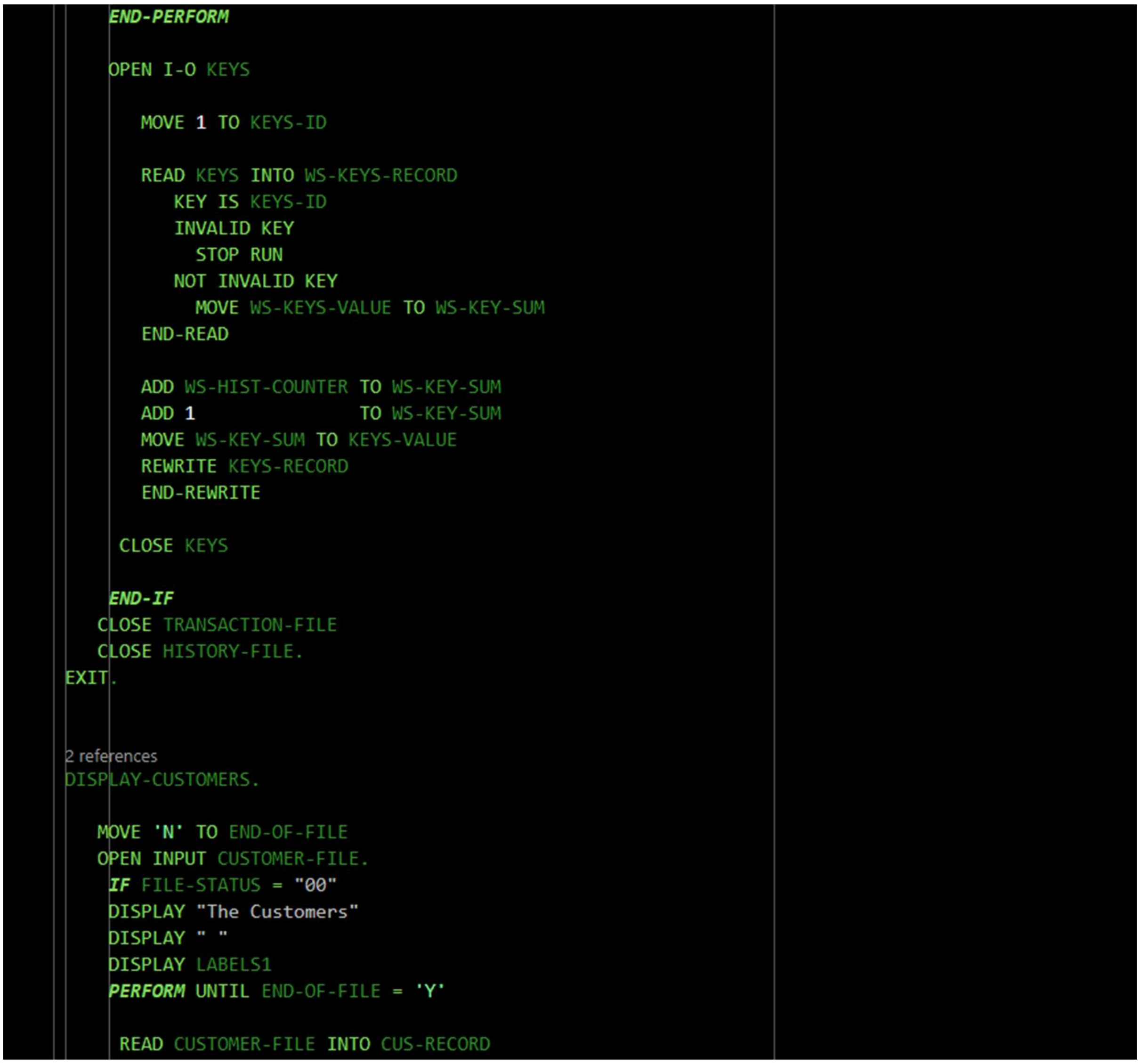
```
 READ HISTORY-FILE INTO WS-HISTORY-RECORD
  AT END
   MOVE 'Y' TO END-OF-FILE

  NOT AT END
  MOVE WS-HIS-ID TO HIS-ID

  MOVE WS-HIS-TRANSACTION-DATE TO INPUT-DATE
  call 'CalculateEpochDays'   USING YEAR-IN MONTH-IN DAY-IN
  TOTALRETURN2

 END-READ

COMPUTE WS-DATERANGE = TOTALRETURN1 - TOTALRETURN2

 IF WS-DATERANGE >= 100 AND WS-HIS-TRANSACTION-DATE >= 1
  DELETE HISTORY-FILE RECORD
   INVALID KEY
     DISPLAY "Invalid Key"
   NOT INVALID KEY
     DISPLAY "Record Deleted"
  END-DELETE
 END-IF

END-PERFORM

ELSE IF FILE-STATUS2 = "35"
DISPLAY "HISTORY-FILE       "
            ": doesn't exist.Please Load the file..."
    OPEN OUTPUT HISTORY-FILE
        CONTINUE
    CLOSE HISTORY-FILE

  ELSE
 DISPLAY "Error opening file. FILE STATUS: " FILE-STATUS2

 END-IF
CLOSE HISTORY-FILE.
EXIT.
```



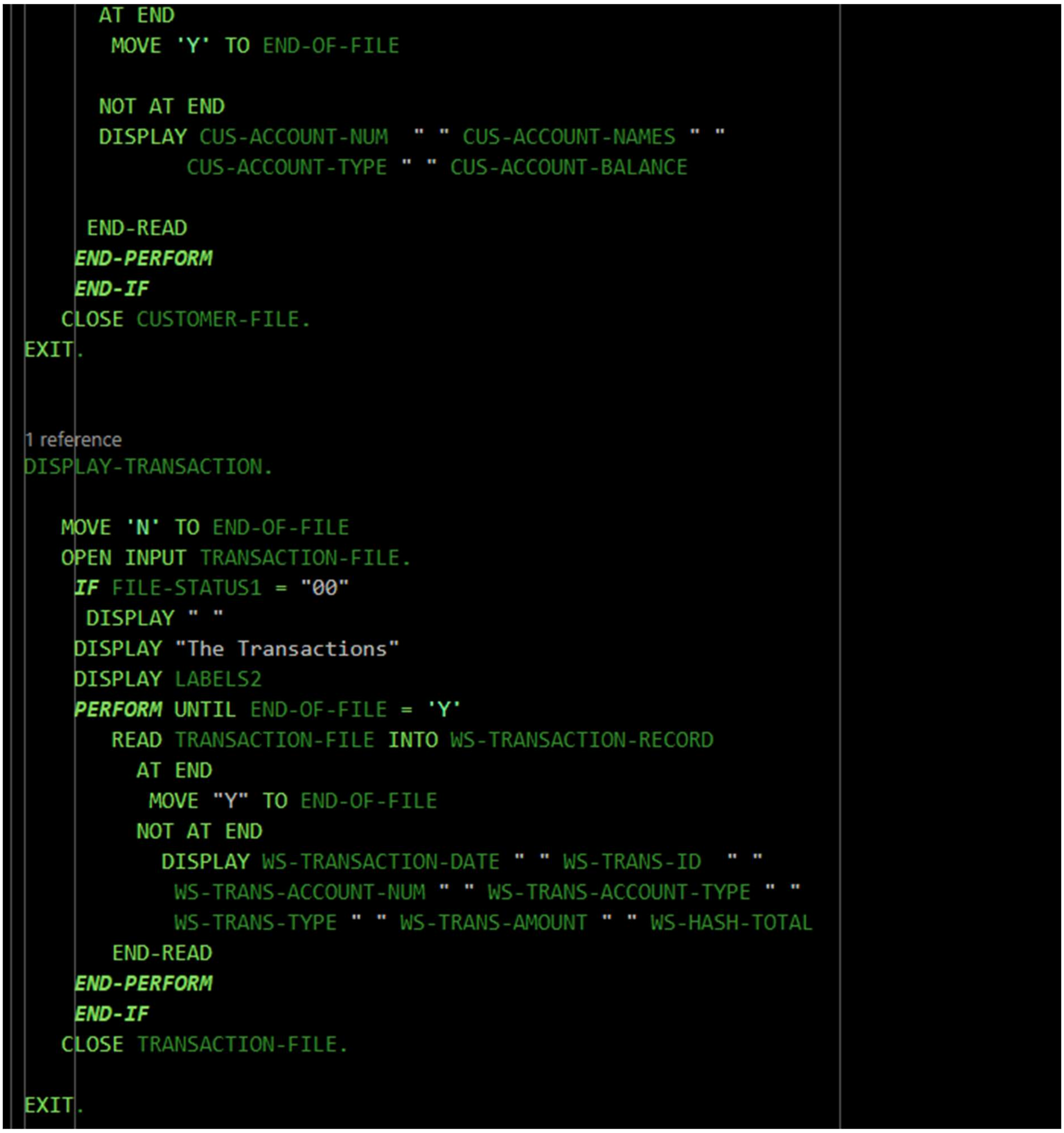
```
TRANSACTION-FILE-CHECK.

  IF FILE-STATUS1 = "00"
    DISPLAY "TRANSACTION-FILE exists and opened successfully."

    OPEN I-O CUSTOMER-FILE
    PERFORM UNTIL END-OF-FILE2 = 'Y'

    READ TRANSACTION-FILE INTO WS-TRANSACTION-RECORD
     AT END
      MOVE "Y" TO END-OF-FILE2
     NOT AT END
      MOVE WS-TRANS-ACCOUNT-NUM TO ACCOUNT-NUM

       READ CUSTOMER-FILE INTO CUS-RECORD KEY IS ACCOUNT-NUM
        INVALID KEY
         DISPLAY "Customer record not found for account: "
        NOT INVALID KEY
          ADD WS-TRANS-AMOUNT TO CUS-ACCOUNT-BALANCE

          MOVE CUS-ACCOUNT-BALANCE TO ACCOUNT-BALANCE
          REWRITE CUSTOMER-RECORD
          ADD 1 TO WS-HASH-TOTAL-TEMP
       END-READ
    END-READ
    END-PERFORM
    DISPLAY "FILE VALIDATION"
    IF WS-HASH-TOTAL = WS-HASH-TOTAL-TEMP
       DISPLAY "The HASH FILES Match"

    ELSE
      DISPLAY "ABEND"
        STOP RUN
    END-IF
    CLOSE CUSTOMER-FILE

    ELSE IF FILE-STATUS1 = "35"
    DISPLAY "TRANSACTION-FILE   "
                ": doesn't exist. Please Load the file..."
   ELSE
    DISPLAY "Error opening file. FILE STATUS: " FILE-STATUS1
```



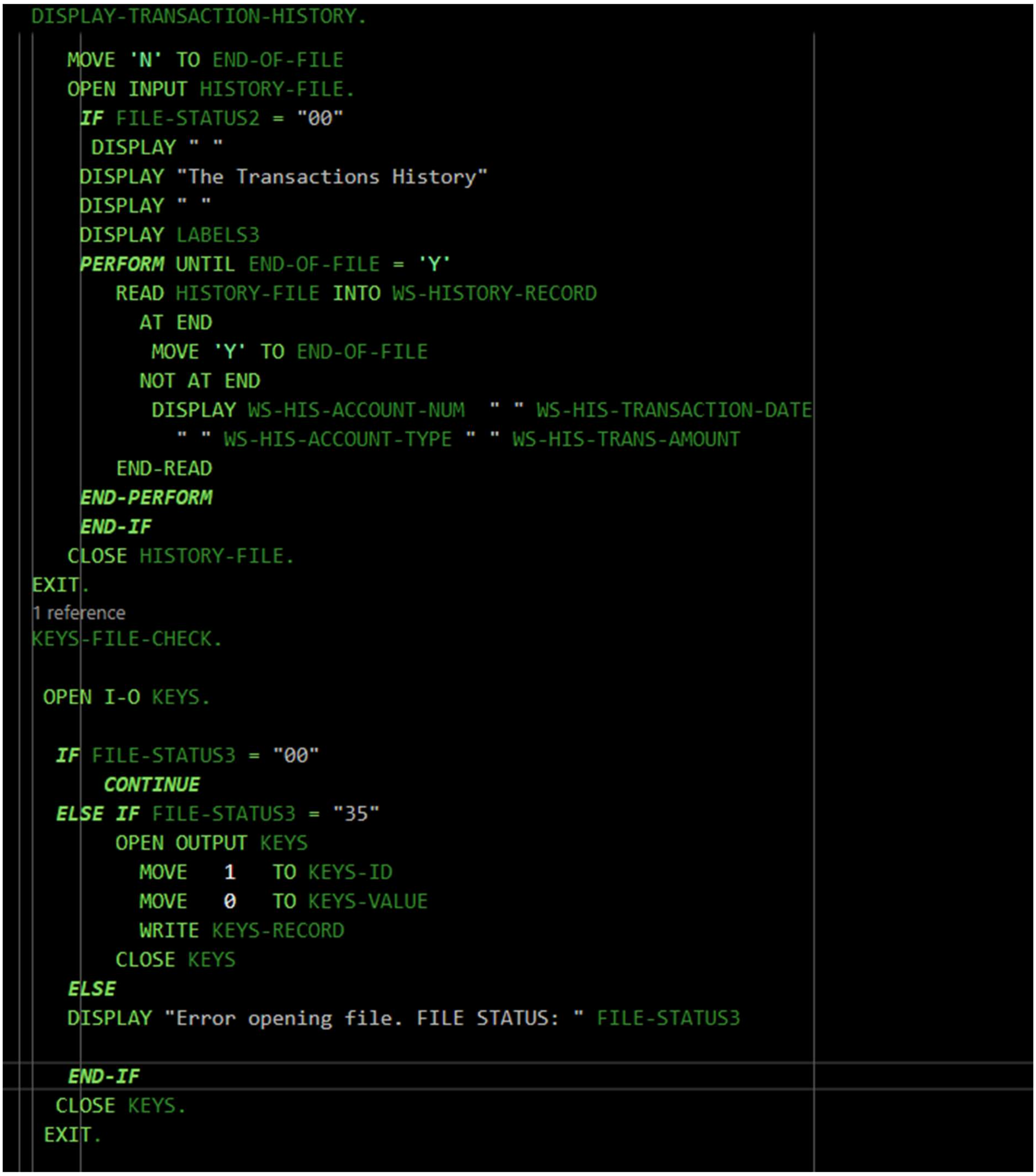
```
 SPACING.
        DISPLAY " "
    END-PERFORM
 EXIT.

1 reference
WRITE-TRANSACTION-HISTORY.

   OPEN I-O HISTORY-FILE.
   MOVE 'N' TO END-OF-FILE
   OPEN INPUT TRANSACTION-FILE
    IF FILE-STATUS1 = "00"
     DISPLAY " "

    OPEN INPUT KEYS
     READ KEYS INTO WS-KEYS-RECORD
        AT END
         CONTINUE
        NOT AT END
          IF WS-KEYS-VALUE > 10
            MOVE WS-KEYS-VALUE TO WS-HIST-COUNTER
          END-IF

     END-READ
    CLOSE KEYS


    PERFORM UNTIL END-OF-FILE = 'Y'
       READ TRANSACTION-FILE INTO WS-TRANSACTION-RECORD
         AT END
          MOVE "Y" TO END-OF-FILE
         NOT AT END
          MOVE WS-HIST-COUNTER        TO HIS-ID
          MOVE WS-TRANSACTION-DATE    TO HIS-TRANSACTION-DATE
          MOVE WS-TRANS-ACCOUNT-NUM   TO HIS-ACCOUNT-NUM
          MOVE WS-TRANS-ACCOUNT-TYPE  TO HIS-ACCOUNT-TYPE
          MOVE WS-TRANS-AMOUNT        TO HIS-TRANS-AMOUNT
          WRITE HISTORY-RECORD
         ADD 1 TO WS-HIST-COUNTER
         ADD 1 TO WS-HIST-COUNTER2
       END-READ
```



```
    END-PERFORM

    OPEN I-O KEYS

       MOVE 1 TO KEYS-ID

       READ KEYS INTO WS-KEYS-RECORD
          KEY IS KEYS-ID
          INVALID KEY
            STOP RUN
          NOT INVALID KEY
            MOVE WS-KEYS-VALUE TO WS-KEY-SUM
       END-READ

       ADD WS-HIST-COUNTER TO WS-KEY-SUM
       ADD 1               TO WS-KEY-SUM
       MOVE WS-KEY-SUM TO KEYS-VALUE
       REWRITE KEYS-RECORD
       END-REWRITE

     CLOSE KEYS

    END-IF
   CLOSE TRANSACTION-FILE
   CLOSE HISTORY-FILE.
EXIT.


2 references
DISPLAY-CUSTOMERS.

   MOVE 'N' TO END-OF-FILE
   OPEN INPUT CUSTOMER-FILE.
    IF FILE-STATUS = "00"
    DISPLAY "The Customers"
    DISPLAY " "
    DISPLAY LABELS1
    PERFORM UNTIL END-OF-FILE = 'Y'

     READ CUSTOMER-FILE INTO CUS-RECORD
```



```
      AT END
       MOVE 'Y' TO END-OF-FILE

      NOT AT END
      DISPLAY CUS-ACCOUNT-NUM  " " CUS-ACCOUNT-NAMES " "
             CUS-ACCOUNT-TYPE " " CUS-ACCOUNT-BALANCE

     END-READ
    END-PERFORM
    END-IF
   CLOSE CUSTOMER-FILE.
EXIT.


1 reference
DISPLAY-TRANSACTION.

   MOVE 'N' TO END-OF-FILE
   OPEN INPUT TRANSACTION-FILE.
    IF FILE-STATUS1 = "00"
     DISPLAY " "
    DISPLAY "The Transactions"
    DISPLAY LABELS2
    PERFORM UNTIL END-OF-FILE = 'Y'
       READ TRANSACTION-FILE INTO WS-TRANSACTION-RECORD
         AT END
          MOVE "Y" TO END-OF-FILE
         NOT AT END
           DISPLAY WS-TRANSACTION-DATE " " WS-TRANS-ID  " "
            WS-TRANS-ACCOUNT-NUM " " WS-TRANS-ACCOUNT-TYPE " "
            WS-TRANS-TYPE " " WS-TRANS-AMOUNT " " WS-HASH-TOTAL
       END-READ
    END-PERFORM
    END-IF
   CLOSE TRANSACTION-FILE.

EXIT.
```



```
DISPLAY-TRANSACTION-HISTORY.

   MOVE 'N' TO END-OF-FILE
   OPEN INPUT HISTORY-FILE.
    IF FILE-STATUS2 = "00"
     DISPLAY " "
    DISPLAY "The Transactions History"
    DISPLAY " "
    DISPLAY LABELS3
    PERFORM UNTIL END-OF-FILE = 'Y'
       READ HISTORY-FILE INTO WS-HISTORY-RECORD
         AT END
          MOVE 'Y' TO END-OF-FILE
         NOT AT END
          DISPLAY WS-HIS-ACCOUNT-NUM  " " WS-HIS-TRANSACTION-DATE
            " " WS-HIS-ACCOUNT-TYPE " " WS-HIS-TRANS-AMOUNT
       END-READ
    END-PERFORM
    END-IF
   CLOSE HISTORY-FILE.
EXIT.
1 reference
KEYS-FILE-CHECK.

 OPEN I-O KEYS.

  IF FILE-STATUS3 = "00"
      CONTINUE
  ELSE IF FILE-STATUS3 = "35"
       OPEN OUTPUT KEYS
         MOVE   1   TO KEYS-ID
         MOVE   0   TO KEYS-VALUE
         WRITE KEYS-RECORD
       CLOSE KEYS
   ELSE
   DISPLAY "Error opening file. FILE STATUS: " FILE-STATUS3

   END-IF
  CLOSE KEYS.
 EXIT.
```