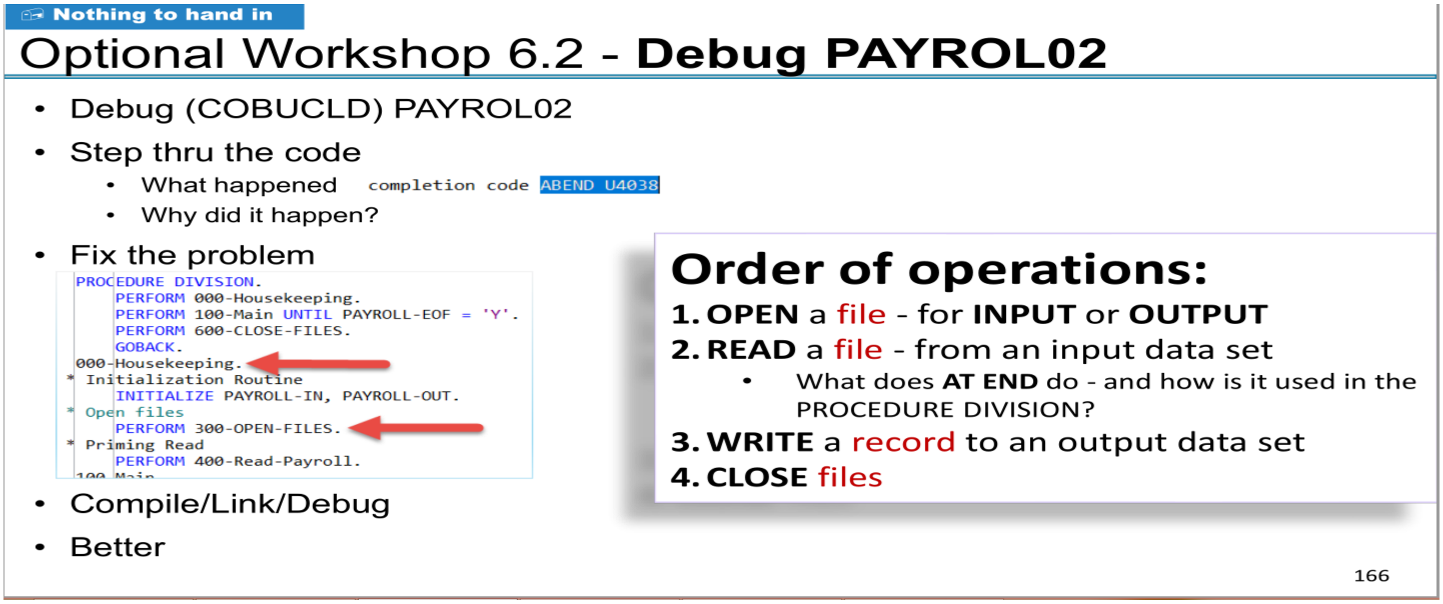


Nothing to hand in

Optional Workshop 6.2 - **Debug PAYROL02**

- Debug (COBUCLD) PAYROL02
- Step thru the code
  - What happened `completion code ABEND U4038`
  - Why did it happen?
- Fix the problem

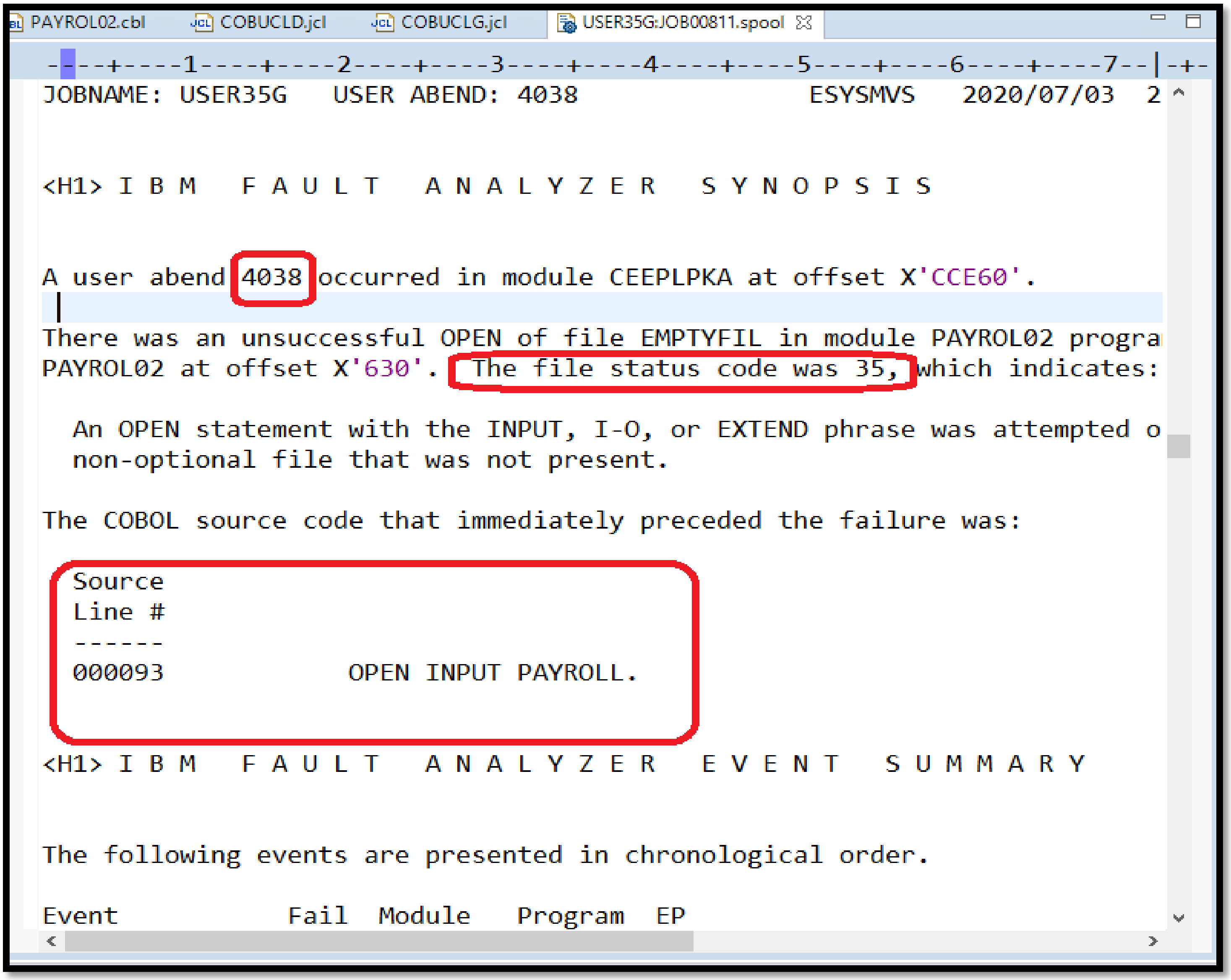
```
PROCEDURE DIVISION.
     PERFORM 000-Housekeeping.
     PERFORM 100-Main UNTIL PAYROLL-EOF = 'Y'.
     PERFORM 600-CLOSE-FILES.
     GOBACK.
000-Housekeeping.
* Initialization Routine
     INITIALIZE PAYROLL-IN, PAYROLL-OUT.
* Open files
     PERFORM 300-OPEN-FILES.
* Priming Read
     PERFORM 400-Read-Payroll.
100-Main.
```

- Compile/Link/Debug
- Better

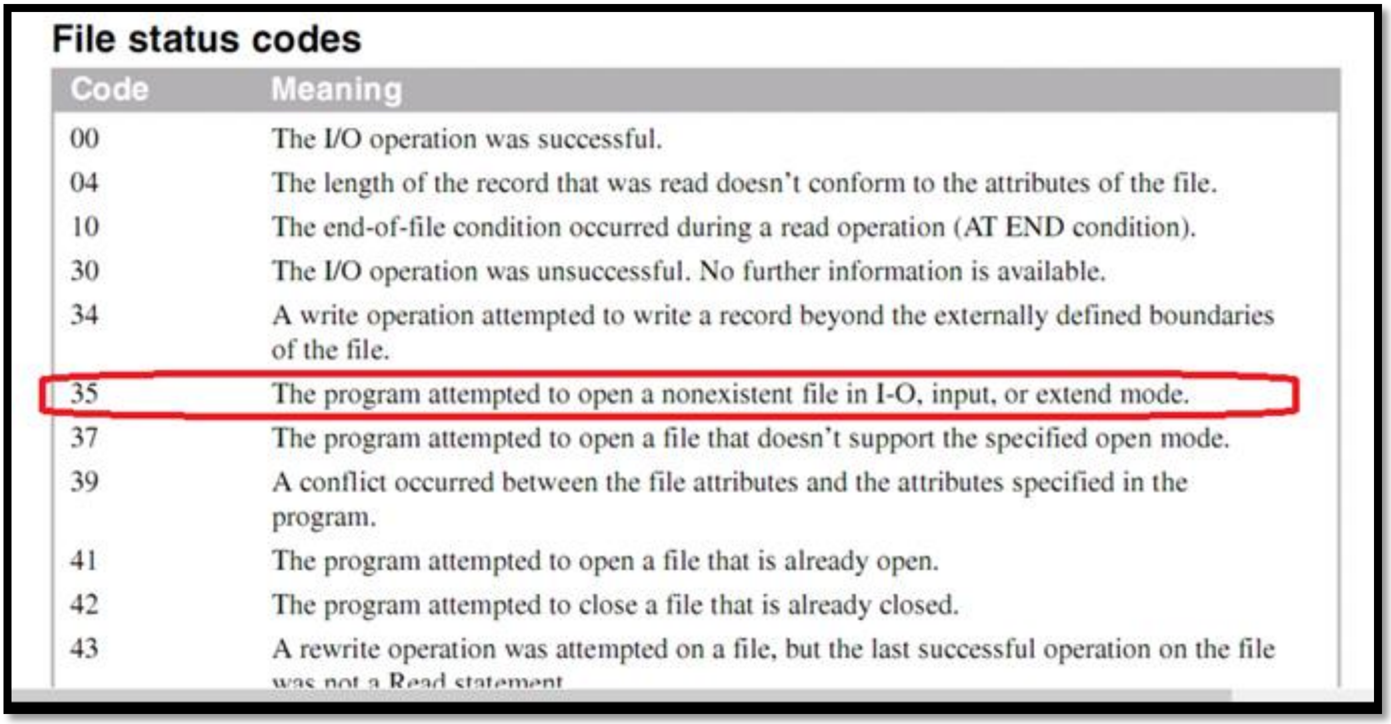
Order of operations:

1. **OPEN** a file - for **INPUT** or **OUTPUT**
2. **READ** a file - from an input data set
   - What does **AT END** do - and how is it used in the PROCEDURE DIVISION?
3. **WRITE** a record to an output data set
4. **CLOSE** files

Solution : empty file / unfound file will lead to abend 4038 .



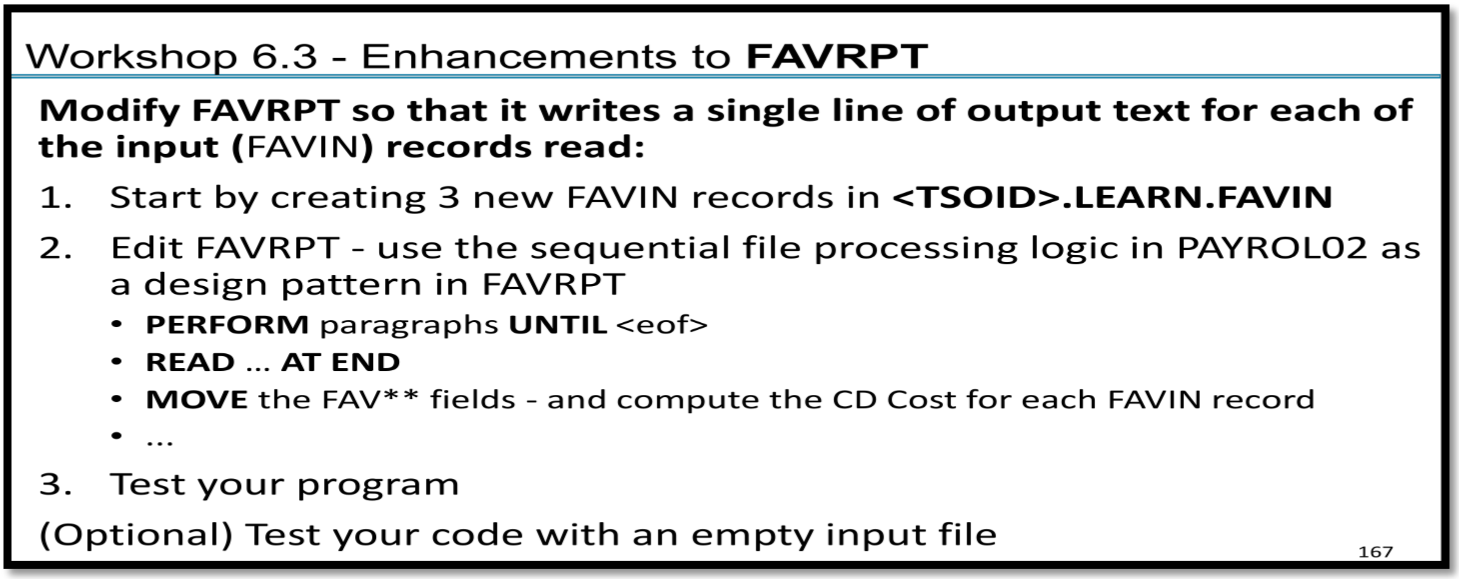
```
 --+----1----+----2----+----3----+----4----+----5----+----6----+----7--|-+-
JOBNAME: USER35G   USER ABEND: 4038              ESYSMVS   2020/07/03   2


<H1> I B M   F A U L T   A N A L Y Z E R   S Y N O P S I S


A user abend 4038 occurred in module CEEPLPKA at offset X'CCE60'.

There was an unsuccessful OPEN of file EMPTYFIL in module PAYROL02 progra
PAYROL02 at offset X'630'.  The file status code was 35, which indicates:

  An OPEN statement with the INPUT, I-O, or EXTEND phrase was attempted o
  non-optional file that was not present.

The COBOL source code that immediately preceded the failure was:

  Source
  Line #
  ------
  000093           OPEN INPUT PAYROLL.


<H1> I B M   F A U L T   A N A L Y Z E R   E V E N T   S U M M A R Y


The following events are presented in chronological order.

Event               Fail   Module    Program   EP
```



File status codes

Code	Meaning
00	The I/O operation was successful.
04	The length of the record that was read doesn't conform to the attributes of the file.
10	The end-of-file condition occurred during a read operation (AT END condition).
30	The I/O operation was unsuccessful. No further information is available.
34	A write operation attempted to write a record beyond the externally defined boundaries of the file.
35	The program attempted to open a nonexistent file in I-O, input, or extend mode.
37	The program attempted to open a file that doesn't support the specified open mode.
39	A conflict occurred between the file attributes and the attributes specified in the program.
41	The program attempted to open a file that is already open.
42	The program attempted to close a file that is already closed.
43	A rewrite operation was attempted on a file, but the last successful operation on the file was not a Read statement

Question



Workshop 6.3 - Enhancements to **FAVRPT**

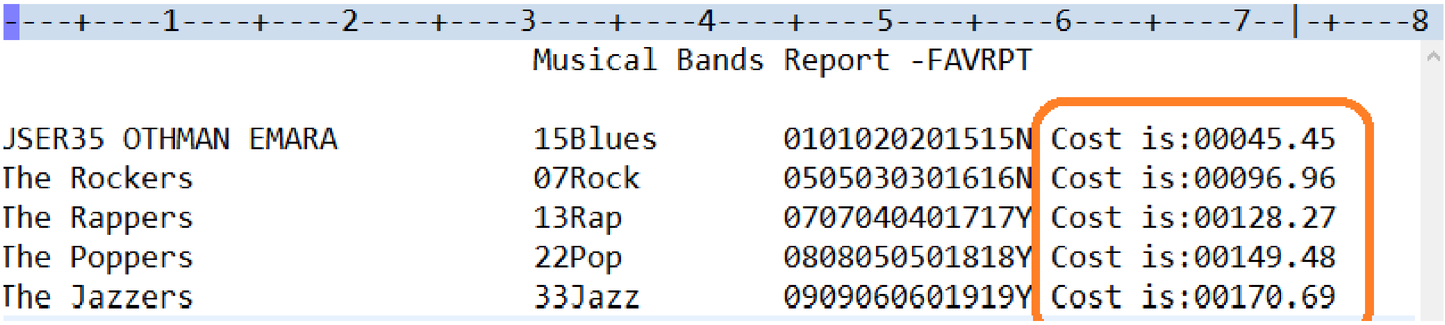
Modify FAVRPT so that it writes a single line of output text for each of the input (FAVIN**) records read:**

1. Start by creating 3 new FAVIN records in **<TSOID>.LEARN.FAVIN**
2. Edit FAVRPT - use the sequential file processing logic in PAYROL02 as a design pattern in FAVRPT
   - **PERFORM** paragraphs **UNTIL** <eof>
   - **READ** … **AT END**
   - **MOVE** the FAV** fields - and compute the CD Cost for each FAVIN record
   - …
3. Test your program

(Optional) Test your code with an empty input file

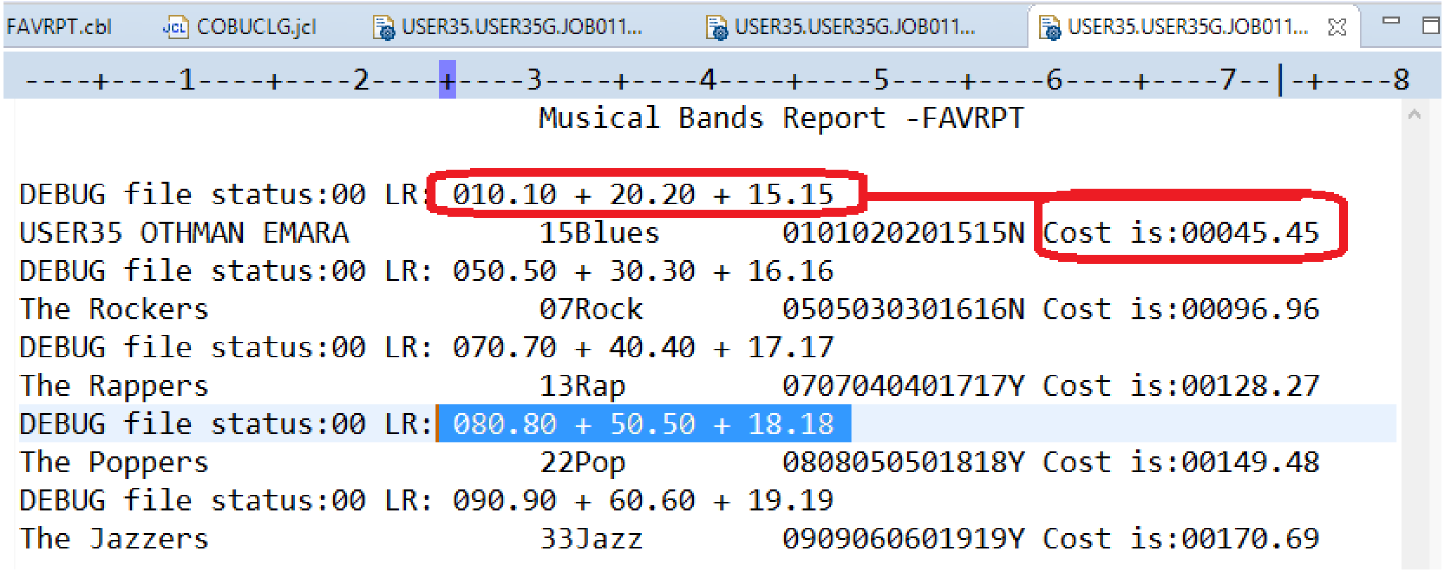
167

Answer : Report done with cost printed .



```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--|-+----8
                              Musical Bands Report -FAVRPT

JSER35 OTHMAN EMARA           15Blues       0101020201515N Cost is:00045.45
The Rockers                   07Rock        0505030301616N Cost is:00096.96
The Rappers                   13Rap         0707040401717Y Cost is:00128.27
The Poppers                   22Pop         0808050501818Y Cost is:00149.48
The Jazzers                   33Jazz        0909060601919Y Cost is:00170.69
```

Code & debug picture below , for some reason the debug script giving JCL error .



```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--|-+----8
                              Musical Bands Report -FAVRPT

DEBUG file status:00 LR: 010.10 + 20.20 + 15.15
USER35 OTHMAN EMARA           15Blues       0101020201515N Cost is:00045.45
DEBUG file status:00 LR: 050.50 + 30.30 + 16.16
The Rockers                   07Rock        0505030301616N Cost is:00096.96
DEBUG file status:00 LR: 070.70 + 40.40 + 17.17
The Rappers                   13Rap         0707040401717Y Cost is:00128.27
DEBUG file status:00 LR: 080.80 + 50.50 + 18.18
The Poppers                   22Pop         0808050501818Y Cost is:00149.48
DEBUG file status:00 LR: 090.90 + 60.60 + 19.19
The Jazzers                   33Jazz        0909060601919Y Cost is:00170.69
```

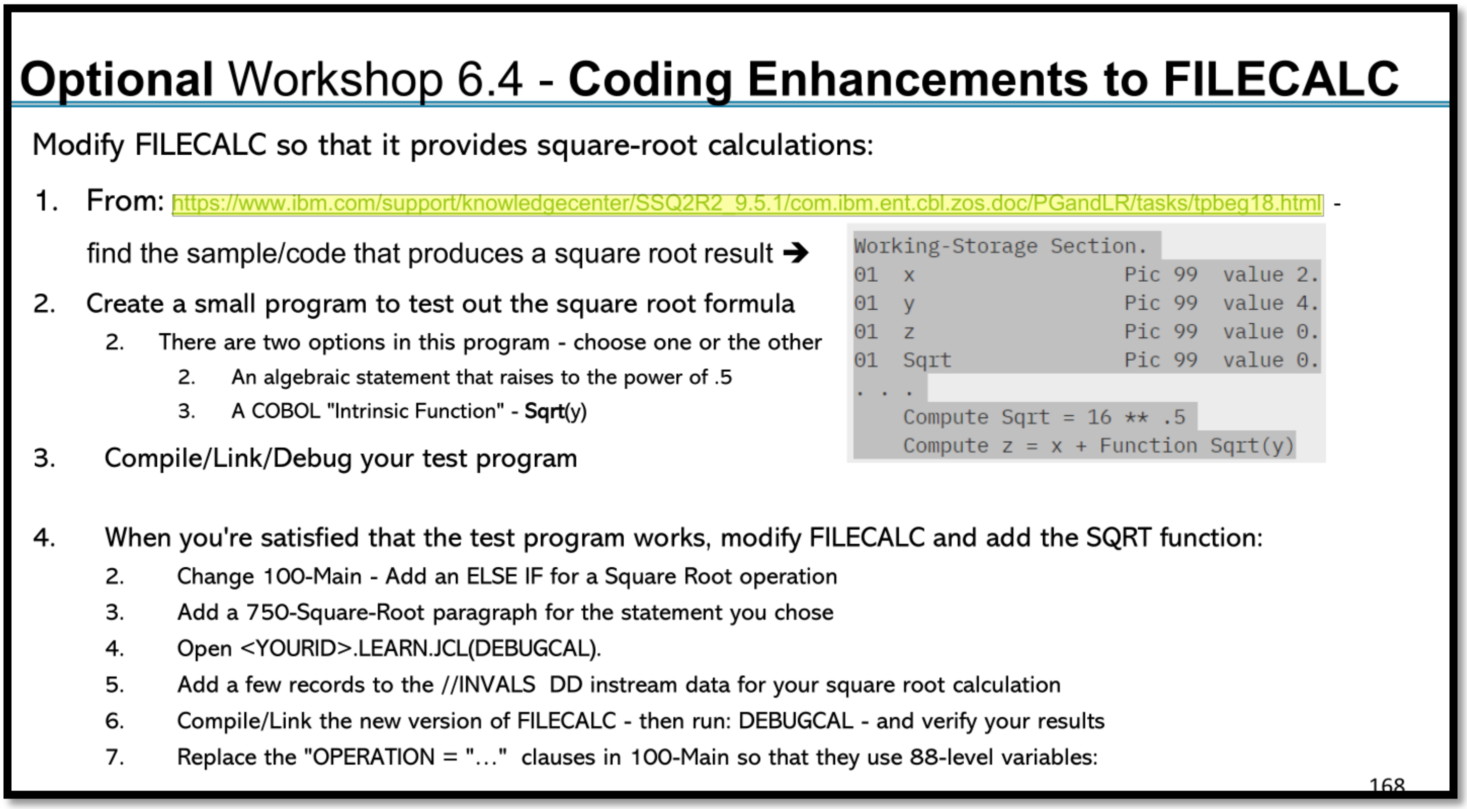
The code :

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. FAVRPT.
      ***** This MODULE Workshop 5.3.1b - Create new COBOL program
      *****  THAT USES A FILE READ
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT FAVIN1  ASSIGN TO FAVIN
                          FILE STATUS IS FAVIN-F-STATUS.
           SELECT OUT2    ASSIGN TO FAVRPT.
       DATA DIVISION.
       FILE SECTION.
       FD  FAVIN1
           RECORDING MODE IS F
           LABEL RECORDS ARE STANDARD
           RECORD CONTAINS 80 CHARACTERS
           BLOCK CONTAINS 0 RECORDS
           DATA RECORD IS FAVIN-REC.
       01 FAVIN-REC.
          05 ARTIST-NAME                PIC X(30).
          05 NUMBER-OF-MUSICIAN         PIC 9(02).
          05 MUSICAL-GENRE              PIC X(12).
          05 COST.
             10 CD-COST                 PIC 9(3)V99.
             10 SHIPPING-COST           PIC 9(2)V99.
             10 TAX                     PIC 9(2)V99.
          05 BAND-IS-STILL-TOGETHER     PIC X(1).
       FD  OUT2
```

```
           RECORDING MODE IS F
           LABEL RECORDS ARE STANDARD
           RECORD CONTAINS 80 CHARACTERS
           BLOCK CONTAINS 0 RECORDS
           DATA RECORD IS FAVOUT-REC.
       01 FAVOUT-REC.
          05 ARTIST-NAME-OUT            PIC X(30).
          05 NUMBER-OF-MUSICIAN-OUT     PIC 9(02).
          05 MUSICAL-GENRE-OUT          PIC X(12).
          05 COST-OUT.
             10 CD-COST-OUT             PIC 9(3)V99.
             10 SHIPPING-COST-OUT       PIC 9(2)V99.
             10 TAX-OUT                 PIC 9(2)V99.
          05 BAND-IS-STILL-TOGETHER-OUT PIC X(1).
          05 COST-IS                    PIC X(9)    VALUE ' Cost is:'.
          05 COMPUTED-COST-OUT          PIC 9(5).99.

       WORKING-STORAGE SECTION.
       01 HEADER-1.
          05 FILLER                     PIC X(30)   VALUE SPACES.
          05 FILLER                     PIC X(30)   VALUE
                                         'Musical Bands Report -
FAVRPT'.
          05 FILLER                     PIC X(20)   VALUE SPACES.
       01 DEBUG-REC.
          05 FILLER                     PIC X(5)    VALUE 'DEBUG'

JUSTIFIED

RIGHT.
          05 FILLER                     PIC X(13)   VALUE
                                                        ' file
status:'.
          05 FAVIN-F-STATUS             PIC X(2).
          05 FILLER                     PIC X(4)    VALUE ' LR:'.
          05 LASTREC                    PIC X       VALUE SPACES.
             88 STELL-THERE-REC                     VALUE ' '.
             88 NO-MORE-RECORDS                     VALUE 'Y'.
          05   DBG-MESSAGE-ALL.
              10 DBG-MESSAGE.
                  15 DBG-MSG1                   PIC X(15).
                  15 DBG-MSG2                   PIC X(15).
              10 DBG-MSG3                   PIC X(10).
```

```
        10 DBG-MSG4                   PIC X(10).
        10 DBG-MSG5                   PIC X(5).
 01 COST-DEBUG.
        10 CD-COST-D                  PIC 9(3).99.
        10 FILLER                     PIC X(3)    VALUE ' + '.
        10 SHIPPING-COST-D            PIC 9(2).99.
        10 FILLER                     PIC X(3)    VALUE ' + '.
        10 TAX-D                      PIC 9(2).99.
 77 COMPUTED-COST                     PIC 9(5).99.
 PROCEDURE DIVISION.
     OPEN INPUT FAVIN1.
     OPEN OUTPUT OUT2.
     WRITE FAVOUT-REC FROM HEADER-1.
     MOVE SPACES TO FAVOUT-REC.
     WRITE FAVOUT-REC AFTER ADVANCING 1 LINES.
*    Prime Read
     PERFORM READ-RECORD.
     PERFORM UNTIL LASTREC = 'Y' OR NO-MORE-RECORDS
             PERFORM PROCESS-RECORDS
             PERFORM WRITE-RECORD
             PERFORM READ-RECORD
     END-PERFORM
     PERFORM CLOSE-FILES
     STOP RUN.
 READ-RECORD.
     READ FAVIN1
*         AT END   MOVE 'Y' TO LASTREC
     AT END
        PERFORM END-OF-FILE
*    NOT AT END         PERFORM PROCESS-RECORDS
     END-READ.
 PROCESS-RECORDS.
     COMPUTE COMPUTED-COST =(CD-COST + SHIPPING-COST + TAX).
     MOVE CD-COST TO CD-COST-D .
     MOVE SHIPPING-COST TO SHIPPING-COST-D .
     MOVE TAX TO TAX-D .
     MOVE SPACES TO DBG-MESSAGE-ALL.
*    MOVE ' P-RCD ' TO DBG-MSG1 .
*    WRITE FAVOUT-REC FROM DEBUG-REC.
*    Just to print debug data
*    MOVE COST-DEBUG TO DBG-MESSAGE .
*    WRITE FAVOUT-REC FROM DEBUG-REC.
 WRITE-RECORD.
```

```
*    MOVE SPACES TO DBG-MESSAGE-ALL.
*    MOVE '  W-RCD ' TO DBG-MSG1 .
*    WRITE FAVOUT-REC FROM DEBUG-REC.
     MOVE FAVIN-REC TO FAVOUT-REC.
     MOVE COMPUTED-COST TO COMPUTED-COST-OUT.
     MOVE ' Cost is:' TO   COST-IS    .
     WRITE FAVOUT-REC.
 CLOSE-FILES.
*    MOVE SPACES TO DBG-MESSAGE-ALL.
*    MOVE ' CLOS-FIL ' TO DBG-MSG1 .
*    WRITE FAVOUT-REC FROM DEBUG-REC.
     CLOSE FAVIN1.
     CLOSE OUT2.
 END-OF-FILE.
     MOVE 'Y' TO LASTREC.
*    MOVE SPACES TO DBG-MESSAGE-ALL.
*    MOVE ' E-O-FILE ' TO DBG-MSG1 .
*    WRITE FAVOUT-REC FROM DEBUG-REC.
```



Optional Workshop 6.4 - **Coding Enhancements to FILECALC**

Modify FILECALC so that it provides square-root calculations:

1. From: https://www.ibm.com/support/knowledgecenter/SSQ2R2_9.5.1/com.ibm.ent.cbl.zos.doc/PGandLR/tasks/tpbeg18.html - find the sample/code that produces a square root result ➔

2. Create a small program to test out the square root formula

   2. There are two options in this program - choose one or the other

      2. An algebraic statement that raises to the power of .5

      3. A COBOL "Intrinsic Function" - **Sqrt**(y)

3. Compile/Link/Debug your test program

```
Working-Storage Section.
01  x                    Pic 99  value 2.
01  y                    Pic 99  value 4.
01  z                    Pic 99  value 0.
01  Sqrt                 Pic 99  value 0.
. . .
    Compute Sqrt = 16 ** .5
    Compute z = x + Function Sqrt(y)
```

4. When you're satisfied that the test program works, modify FILECALC and add the SQRT function:

   2. Change 100-Main - Add an ELSE IF for a Square Root operation

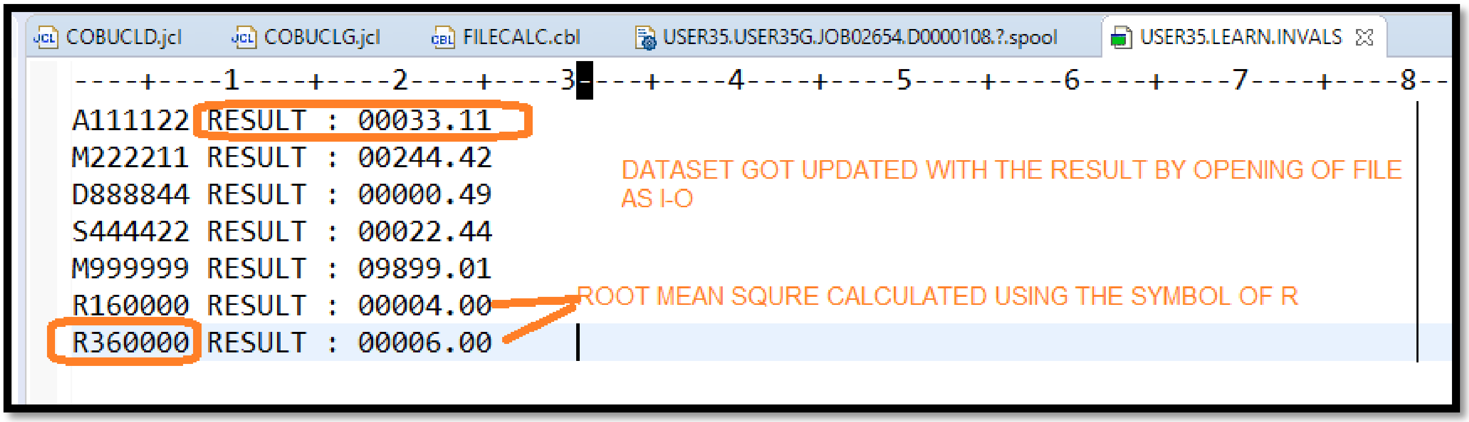
   3. Add a 750-Square-Root paragraph for the statement you chose

   4. Open <YOURID>.LEARN.JCL(DEBUGCAL).

   5. Add a few records to the //INVALS DD instream data for your square root calculation

   6. Compile/Link the new version of FILECALC - then run: DEBUGCAL - and verify your results

   7. Replace the "OPERATION = "…" clauses in 100-Main so that they use 88-level variables:

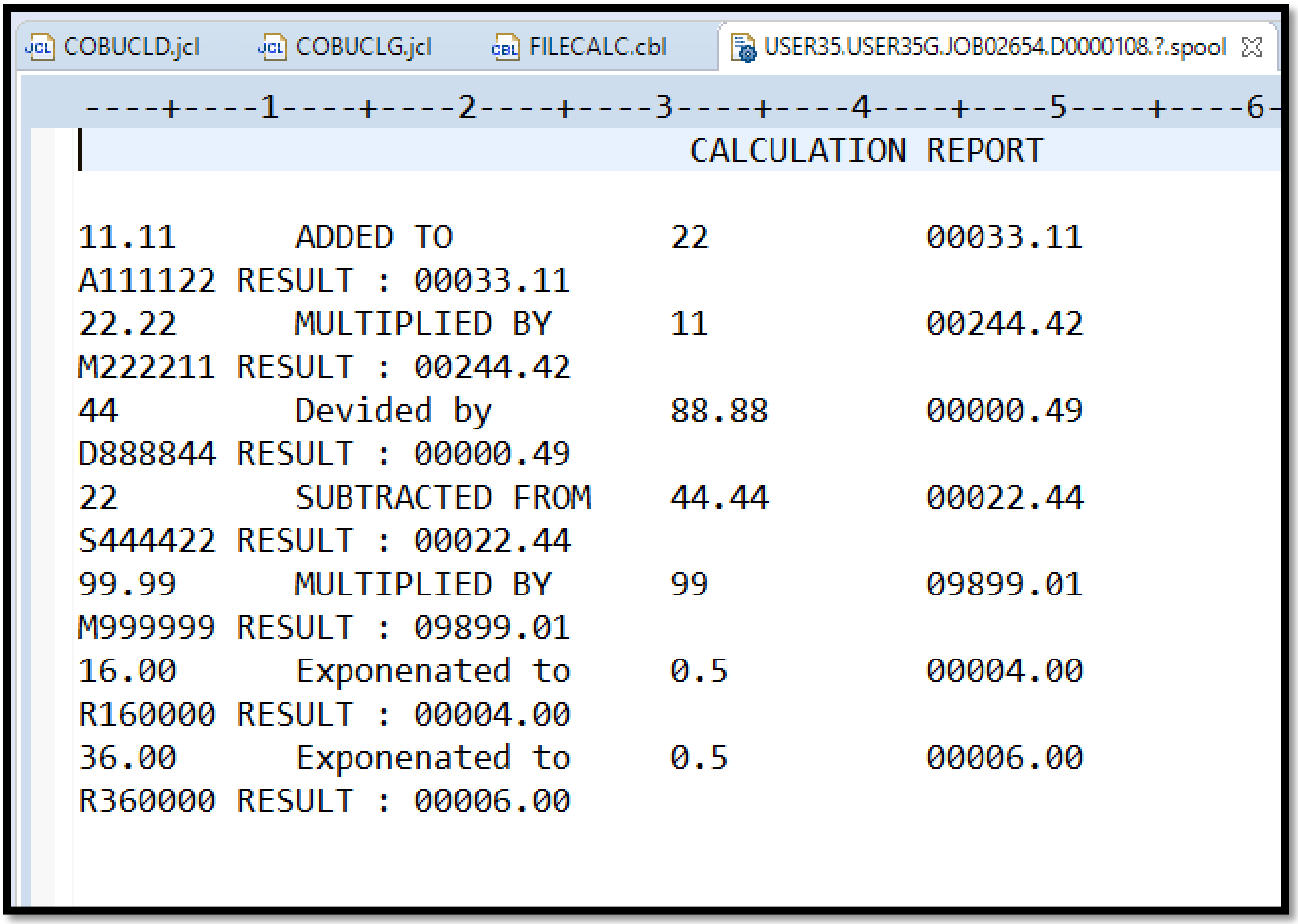


```
A111122 RESULT : 00033.11
M222211 RESULT : 00244.42
D888844 RESULT : 00000.49
S444422 RESULT : 00022.44
M999999 RESULT : 09899.01
R160000 RESULT : 00004.00
R360000 RESULT : 00006.00
```

DATASET GOT UPDATED WITH THE RESULT BY OPENING OF FILE AS I-O

ROOT MEAN SQURE CALCULATED USING THE SYMBOL OF R

Also a report generated :



```
                           CALCULATION REPORT

11.11      ADDED TO         22           00033.11
A111122 RESULT : 00033.11
22.22      MULTIPLIED BY    11           00244.42
M222211 RESULT : 00244.42
44         Devided by       88.88        00000.49
D888844 RESULT : 00000.49
22         SUBTRACTED FROM  44.44        00022.44
S444422 RESULT : 00022.44
99.99      MULTIPLIED BY    99           09899.01
M999999 RESULT : 09899.01
16.00      Exponenated to   0.5          00004.00
R160000 RESULT : 00004.00
36.00      Exponenated to   0.5          00006.00
R360000 RESULT : 00006.00
```

Code is :

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. FILECALC.
      * This program reads a file of input values into INVALS-WS
      * The operation read into the W-S structure drives the
arithmetic
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT INVALS ASSIGN TO INVALS ORGANIZATION IS SEQUENTIAL.
           SELECT REPORTFILE ASSIGN TO PRTLINE.
```

```
 DATA DIVISION.
 FILE SECTION.
 FD  INVALS
     RECORDING MODE IS F
     LABEL RECORDS ARE STANDARD
     RECORD CONTAINS 80 CHARACTERS
     BLOCK CONTAINS 0 RECORDS
     DATA RECORD IS INVALS-REC.
 01  INVALS-REC  PIC X(80).
  FD REPORTFILE
     RECORDING MODE IS F
     LABEL RECORDS ARE STANDARD
     RECORD CONTAINS 80 CHARACTERS
     BLOCK CONTAINS 0 RECORDS
     DATA RECORD IS REPORT-REC.
 01 REPORT-REC .
     05 VAL1-RPT     PIC X(10) VALUE SPACES .
     05 OPERATION-CODE    PIC X(20) VALUE SPACES .
     05 VAL2-RPT   PIC X(10) VALUE SPACES .
     05 EQUAL-TEXT    PIC X(3) VALUE ' = ' .
     05 RESULT-RPT   PIC  9(5).99  VALUE ZEROS .
     05 FILLER   PIC X(20) VALUE SPACES  .
 WORKING-STORAGE SECTION.
* End of File switch
 01  INVALS-EOF               PIC X(1) VALUE SPACE.
      88   EOF-REACHED        VALUE   'Y'.
 01  INVALS-WS.
     05  OPERATION  PIC X(1) .
         88  ADD-OPERATION            VALUE  'A' 'a'.
         88  SUBTRACT-OPERATION       VALUE  'S'  's'.
         88  MULTIPLY-OPERATION       VALUE  'M' 'm'.
         88  SQUARE-ROOT-OPERATION    VALUE  'R'  'r'.
         88  DIVIDE-OPERATION         VALUE  'D' 'd'.
     05  INVALS-1                PIC 99V99.
     05  INVALS-2                PIC 99.
     05  INVALS-2X REDEFINES INVALS-2  PIC X(2).
     05  TEXT1                   PIC X(10) VALUE ' RESULT : ' .
     05  INVALS-RZLT             PIC 99999.99.
 01 HEADER-1.
    05 FILLER    PIC X(30)   VALUE SPACES.
    05 FILLER    PIC X(30)   VALUE  ' CALCULATION REPORT '.
 01  INVALS1-EDITED     PIC 99.99.
 01  INVALS-1X REDEFINES INVALS1-EDITED  PIC X(5).
```

```
       PROCEDURE DIVISION.
           PERFORM 000-Housekeeping.
           PERFORM 100-Main UNTIL EOF-REACHED.
           PERFORM 900-CLOSE-FILES.
           GOBACK.
       000-Housekeeping.
           INITIALIZE INVALS-WS.
           PERFORM 300-OPEN-FILES.
           WRITE REPORT-REC FROM HEADER-1.
           MOVE SPACES TO REPORT-REC.
           WRITE REPORT-REC AFTER ADVANCING 1 LINES.
      *    Priming Read
           PERFORM 400-Read-INVALS.
       100-Main.
           PERFORM 320-PROCESS-RECORDS
           PERFORM 350-REWRITE-RECORD
           PERFORM 400-Read-INVALS .
       300-OPEN-FILES.
           OPEN I-O INVALS.
           OPEN OUTPUT REPORTFILE.
       320-PROCESS-RECORDS.
           IF ADD-OPERATION PERFORM 500-ADD
           ELSE IF SUBTRACT-OPERATION PERFORM 600-SUBTRACT
           ELSE IF MULTIPLY-OPERATION PERFORM 700-MULTIPLY
           ELSE IF SQUARE-ROOT-OPERATION PERFORM 750-SQUARE-ROOT
           ELSE IF DIVIDE-OPERATION PERFORM 800-DIVIDE
           END-IF.
      *    DISPLAY INVALS-1.
      *    DISPLAY INVALS-2.
      *    DISPLAY INVALS-RESULT.
       350-REWRITE-RECORD.
      *    Although updating with calculated field is redundancy
             STRING    INVALS-1X
               DELIMITED BY SIZE INTO VAL1-RPT .
              STRING  INVALS-2X
                 DELIMITED BY SIZE INTO  VAL2-RPT.
            IF SQUARE-ROOT-OPERATION
                MOVE '0.5' TO  VAL2-RPT
            ELSE IF    SUBTRACT-OPERATION OR  DIVIDE-OPERATION
      *  just for Report formt Switching values due to  subtracting x
from y
                 MOVE INVALS-2X  TO  VAL1-RPT
                 MOVE  INVALS-1X  TO  VAL2-RPT
              END-IF.
            MOVE    INVALS-RZLT   TO RESULT-RPT .
            MOVE    ' RESULT : '  TO TEXT1 .
            WRITE REPORT-REC
            WRITE REPORT-REC FROM INVALS-WS.
            REWRITE INVALS-REC FROM INVALS-WS.
       400-Read-INVALS.
           READ INVALS INTO INVALS-WS  AT END MOVE "Y" TO INVALS-EOF
           DISPLAY  INVALS-WS .
           MOVE ZEROS TO INVALS-RZLT  .
           DISPLAY  INVALS-WS .
           MOVE FUNCTION UPPER-CASE(OPERATION) TO OPERATION.
           INITIALIZE REPORT-REC.
           COMPUTE INVALS1-EDITED =  INVALS-1  .
       500-ADD.
           ADD  INVALS-1 , INVALS-2 GIVING INVALS-RZLT.
           MOVE   ' ADDED TO ' TO OPERATION-CODE  .
       600-SUBTRACT.
           SUBTRACT INVALS-2 FROM  INVALS-1 GIVING INVALS-RZLT  .
           MOVE   ' SUBTRACTED FROM ' TO OPERATION-CODE  .
       700-MULTIPLY.
           MULTIPLY  INVALS-1  BY INVALS-2 GIVING INVALS-RZLT  .
           MOVE   ' MULTIPLIED BY ' TO OPERATION-CODE  .
       750-SQUARE-ROOT.
           COMPUTE INVALS-RZLT   =    FUNCTION SQRT( INVALS-1 ) .
           MOVE   ' Exponenated to ' TO OPERATION-CODE  .
       800-DIVIDE.
           DIVIDE INVALS-2 BY INVALS-1    GIVING INVALS-RZLT  .
           MOVE   ' Devided by ' TO OPERATION-CODE  .
       900-CLOSE-FILES.
           CLOSE INVALS.
           CLOSE REPORTFILE.
```