RPGLE (Report Program Generator with Integrated Language Environment)

1. Arithmetic operations program using IF

```
0001.00 DNUM1             S              10  0
0002.00 DNUM2             S              10  0
0003.00 DRESULT           S              10  0
0004.00 Doperation        S               1
0005.00 Dmessage          S              25
0006.00 C     *entry        plist
0007.00 C                   parm                    operation
0008.00 C                   EVAL      NUM1 = 10
0009.00 C                   EVAL      NUM2 = 20
0010.00 C                   IF        operation <> '+' AND operation <> '-'
0011.00 C                             AND operation <> '*' AND operation <> '/'
0012.00 C                   EVAL      message = 'INVALID OPERATOR'
0013.00 C                   ELSE
0014.00 C                   IF        operation = '+'
0015.00 C                   EVAL      RESULT = NUM1 + NUM2
0016.00 C                   EVAL      message = 'Addition = ' + %CHAR(RESULT)
0017.00 C                   ENDIF
0018.00 C                   IF        operation = '-'
0019.00 C                   EVAL      RESULT = NUM1 - NUM2
```

```
0020.00 C                   EVAL      message = 'Subtraction = ' + %CHAR(RESULT)
0021.00 C                   ENDIF
0022.00 C                   IF        operation = '*'
0023.00 C                   EVAL      RESULT = NUM1 * NUM2
0024.00 C                   EVAL      message = 'Multiplication = ' + %CHAR(RESULT)
0025.00 C                   ENDIF
0026.00 C                   IF        operation = '/'
0027.00 C                   EVAL      RESULT = NUM1 / NUM2
0028.00 C                   EVAL      message = 'Division = ' + %CHAR(RESULT)
0029.00 C                   ENDIF
0030.00 C                   ENDIF
0031.00 C     message       DSPLY
0032.00 C                   EVAL      *INLR = '1'
```

2. Arithmetic operations using IF-ELSE

```
0001.00 DNUM1              S              10  0
0002.00 DNUM2              S              10  0
0003.00 DRESULT            S              10  0
0004.00 Doperation         S               1
0005.00 Dmessage           S              25
0006.00 C      *entry        plist
0007.00 C                    parm                    operation
0008.00 C                    EVAL       NUM1 = 10
0009.00 C                    EVAL       NUM2 = 20
0010.00 C                    IF         operation <> '+' AND operation <> '-'
0011.00 C                               AND operation <> '*' AND operation <> '/'
0012.00 C                    EVAL       message = 'INVALID OPERATOR'
0013.00 C                    ELSE
0014.00 C                    IF         operation = '+'
0015.00 C                    EVAL       RESULT = NUM1 + NUM2
0016.00 C                    EVAL       message = 'Addition = ' + %CHAR(RESULT)
0017.00 C                    ELSEIF     operation = '-'
0018.00 C                    EVAL       RESULT = NUM1 - NUM2
0019.00 C                    EVAL       message = 'Subtraction = ' + %CHAR(RESULT)
```

```
0020.00 C                    ELSEIF     operation = '*'
0021.00 C                    EVAL       RESULT = NUM1 * NUM2
0022.00 C                    EVAL       message = 'Multiplication = ' + %CHAR(RESULT)
0023.00 C                    ELSEIF     operation = '/'
0024.00 C                    EVAL       RESULT = NUM1 / NUM2
0025.00 C                    EVAL       message = 'Division = ' + %CHAR(RESULT)
0026.00 C                    ENDIF
0027.00 C                    ENDIF
0028.00 C      message       DSPLY
0029.00 C                    EVAL       *INLR = '1'
```

3. Arithmetic operations using SELECT WHEN

```
0001.00 DNUM1              S              10  0
0002.00 DNUM2              S              10  0
0003.00 DRESULT            S              10  0
0004.00 Doperation         S               1
0005.00 Dmessage           S              25
0006.00 C      *entry        plist
0007.00 C                    parm                    operation
0008.00 C                    EVAL       NUM1 = 10
0009.00 C                    EVAL       NUM2 = 20
0010.00  *
0011.00 C                    SELECT
0012.00 C                    WHEN       operation = '+'
0013.00 C                    EVAL       RESULT = NUM1 + NUM2
0014.00 C                    EVAL       message = 'Addition = ' + %CHAR(RESULT)
0015.00 C                    WHEN       operation = '-'
0016.00 C                    EVAL       RESULT = NUM1 - NUM2
0017.00 C                    EVAL       message = 'Subtraction = ' + %CHAR(RESULT)
0018.00 C                    WHEN       operation = '*'
0019.00 C                    EVAL       RESULT = NUM1 * NUM2
```

```
0020.00 C                   EVAL      message = 'Multiplication = ' + %CHAR(RESULT)
0021.00 C                   WHEN      operation = '/'
0022.00 C                   EVAL      RESULT = NUM1 / NUM2
0023.00 C                   EVAL      message = 'Division = ' + %CHAR(RESULT)
0024.00 C                   OTHER
0025.00 C                   EVAL      message = 'INVALID OPERATOR'
0026.00 C                   ENDSL
0027.00 C     message       DSPLY
0028.00 C                   EVAL      *INLR = '1'
```

4. String concatenation program

```
0001.00 DFNAME              S                10
0002.00 DLNAME              S                10
0003.00 DFULLNAME           S                20
0004.00 C     *entry          plist
0005.00 C                     parm                    FNAME
0006.00 C                     parm                    LNAME
0007.00 C                     EVAL      FULLNAME = %TRIM(FNAME)+' '+%TRIM(LNAME)
0008.00 C     FULLNAME        DSPLY
0009.00 C                     EVAL      *INLR = '1'
```

5. Reading file

```
0001.00 FEMPPF     IF   E           K DISK
0002.00  *
0003.00 C                   READ      EMPPF
0004.00  *
0005.00 C                   DOW       NOT %EOF()
0006.00 C     EMPNAME       DSPLY
0007.00 C                   READ      EMPPF
0008.00 C                   ENDDO
0009.00  *
0010.00 C                   EVAL      *INLR = '1'
```

6. Reading file using SETLL

```
0001.00 FEMPPF     IF   E           K DISK
0002.00  *
0003.00 C                   READ      EMPPF
0004.00  *
0005.00 C                   DOW       NOT %EOF()
0006.00 C     EMPNAME       DSPLY
0007.00 C                   READ      EMPPF
0008.00 C                   ENDDO
0009.00  *
0010.00 C     *LOVAL        SETLL     EMPPF
0011.00 C                   READ      EMPPF
0012.00  *
0013.00 C                   DOW       NOT %EOF()
0014.00 C     EMPNAME       DSPLY
0015.00 C                   READ      EMPPF
0016.00 C                   ENDDO
0017.00  *
0018.00 C                   EVAL      *INLR = '1'
```

7. Reading file using SETGT

```
0010.00 C     *HIVAL        SETGT     EMPPF
0011.00 C                   READP     EMPPF
0012.00  *
0013.00 C                   DOW       NOT %EOF()
0014.00 C     EMPNAME       DSPLY
0015.00 C                   READP     EMPPF
0016.00 C                   ENDDO
0017.00  *
0018.00 C                   EVAL      *INLR = '1'
```

8. SETGT and READP

```
0001.00 FTRANLF    IF   E           K disk
0002.00 Dcnt             S              10  0
0003.00 Dmsg             S              52
0004.00 Dnum             S              10  0
0005.00  *
0006.00 C     *ENTRY       PLIST
0007.00 C                  PARM                    num
0008.00 C                  EVAL      cnt = 0
0009.00 C     *HIVAL       SETGT     TRANLF
0010.00 C                  READP     TRANLF
0011.00 C                  DOW       NOT %EOF() AND cnt < 15
0012.00 C                  if        num = ACCNO
0013.00 C                  EVAL      msg ='AccountNo.:'+%char(ACCNO)+' , '+
0014.00 C                            'TransactionNo.:'+ %char(TRANNO)+' , '
0015.00 C                            + 'TransactionType:'+ TRANTYPE+' , '
0016.00 C                            +'Transaction Amount: '+%char(TRANAMOUNT)
0017.00 C     msg          DSPLY
0018.00 C                  EVAL      cnt = cnt+1
0019.00 C                  ENDIF
```

```
0020.00 C                   READP     TRANLF
0021.00 C                   ENDDO
0022.00  *
0023.00 C                   EVAL      *INLR = '1'
```

9. LEAVE and READPE

```
0001.00 FTRANLF    IF   E           K DISK
0002.00  *
0003.00 Daccnum          S             14  0
0004.00 Dcount           S              3  0
0005.00 Dmsg             S             50
0006.00  *
0007.00 C      *ENTRY       PLIST
0008.00 C                   PARM                    accnum
0009.00  *
0010.00 C                   EVAL      count = 0
0011.00 C      accnum       SETGT     TRANLF
0012.00 C      accnum       READPE    TRANLF
0013.00  *
0014.00  *
0015.00 C                   DOW       NOT %EOF()
0016.00 C                   IF        count = 10
0017.00 C                   LEAVE
0018.00 C                   ENDIF
```

```
0020.00 C      accnum       READPE    TRANLF
0021.00 C                   EVAL      msg = %CHAR(ACCNO)+' '+%CHAR(TRANNO)
0022.00 C                             +' '+TRANTYPE+' '+%CHAR(TRANAMOUNT)
0023.00  *
0024.00 C      msg          DSPLY
0025.00 C                   EVAL      count = count + 1
0026.00  *
0027.00 C                   ENDDO
0028.00  *
0029.00  *
0030.00 C                   EVAL      *INLR = '1'
        ****************** End of data ********************************
```

10. READE

```
0001.00 FEMPLF2    IF   E           K DISK
0002.00  *
0003.00 Dcity            S             20
0004.00 Dmsg             S             50
0005.00  *
0006.00 C      *ENTRY       PLIST
0007.00 C                   PARM                    city
0008.00  *
0009.00 C      city         SETLL     EMPLF2
0010.00 C      city         READE     EMPLF2
0011.00  *
0012.00  *
0013.00 C                   DOW       NOT %EOF()
0014.00  *
0015.00 C                   EVAL      msg = %CHAR(EMPNO)+' '+EMPNAME
0016.00 C                             +' '+EMPCITY+' '+%CHAR(DEPTNO)
0017.00  *
0018.00 C      msg          DSPLY
0019.00 C      city         READE     EMPLF2
```

```
0020.00  *
0021.00 C                   ENDDO
0022.00  *
0023.00  *
0024.00 C                   EVAL       *INLR = '1'
```

11. Employee city count

```
0001.00 FEMPLF2    IF   E           K DISK
0002.00 DMSG              S              20
0002.01 DCOUNT            S               3  0
0003.00 DCURCITY          S              20
0004.00 DPRECITY          S              20
0005.00  *
0006.00 C     *LOVAL        SETLL     EMPLF2
0007.00 C                   READ      EMPREC
0008.00 C                   EVAL      PRECITY = EMPCITY
0009.00 C                   EVAL      COUNT = 1
0010.00  *
0011.00 C                   DOW       NOT %EOF()
0012.00 C                   READ      EMPREC
0013.00 C                   EVAL      CURCITY = EMPCITY
0014.00 C                   IF        PRECITY = CURCITY
0015.00 C                   EVAL      COUNT = COUNT + 1
0016.00 C                   ELSE
0017.00 C                   EVAL      MSG = %TRIM(PRECITY) +' '+%CHAR(COUNT)
0018.00 C     MSG           DSPLY
```

```
0019.00 C                   EVAL      COUNT = 1
0020.00 C                   EVAL      PRECITY = CURCITY
0021.00 C                   ENDIF
0022.00 C                   ENDDO
0023.00  *
0025.00 C                   EVAL      MSG =%TRIM(PRECITY) +' '+%CHAR(COUNT)
0026.00 C     MSG           DSPLY
0027.00 C                   EVAL      *INLR = '1'
```

12. Program using chain

```
0001.00 FEMPLF3    IF   E           K DISK
0002.00 Dempnumber        S              10  0
0003.00 Dmsg              S              20
0004.00  *
0005.00 C     *ENTRY        PLIST
0006.00 C                   PARM                    empnumber
0007.00  *
0008.00 C     empnumber     CHAIN     EMPLF3
0009.00 C                   IF        %FOUND(EMPLF3)
0010.00 C                   EVAL      msg = EMPNAME
0011.00 C     msg           DSPLY
0012.00 C                   ELSE
0013.00 C                   EVAL      msg = 'Not found'
0014.00 C     msg           DSPLY
0015.00 C                   ENDIF
0016.00  *
0017.00 C                   EVAL      *INLR = '1'
```

13. Write and update file

```
0001.00 FEMPPF     UF A E           K DISK
0002.00  *
0003.00 Dempnum           S             10  0
0004.00 Ddeptnum          S             10  0
0005.00 Dcity             S             20
0006.00 Demplname         S             20
0007.00  *
0008.00 C     *ENTRY        PLIST
0009.00 C                   PARM                    empnum
0010.00 C                   PARM                    emplname
0011.00 C                   PARM                    city
0012.00 C                   PARM                    deptnum
0013.00  *
0014.00 C     empnum        CHAIN     EMPPF
0015.00 C                   IF        %FOUND()
0016.00 C                   EVAL      DEPTNO = deptnum
0017.00 C                   UPDATE    EMPREC
0018.00 C                   ELSE
0019.00 C                   EVAL      EMPNO = empnum
```

```
0020.00 C                   EVAL      EMPNAME = emplname
0021.00 C                   EVAL      EMPCITY = city
0022.00 C                   EVAL      DEPTNO = deptnum
0023.00 C                   WRITE     EMPREC
0024.00 C                   ENDIF
0025.00  *
0026.00 C                   EVAL      *INLR = '1'
        ****************** End of data ******************
```

14. Menu program

```
0001.00 FBOOKFILE  UF A E           K DISK
0002.00 FBOOKMENU  CF   E             WORKSTN
0003.00  *
0004.00  *-------------------------------------------
0005.00  *MAIN FILE EXECUTION
0006.00  *-------------------------------------------
0007.00  *
0008.00 C                   DOW       *IN03 = '0'
0009.00 C                   EXFMT     MENUPG
0009.01 C                   EVAL      *IN87 = *OFF
0010.00 C                   EVAL      MENUMSG = ' '
0011.00  *
0012.00 C                   IF        *IN03 = '1'
0013.00 C                   LEAVE
0014.00 C                   ENDIF
0015.00  *
0016.00 C                   IF        *IN06 = '1'
0017.00 C                   EXSR      ENTRYSR
0018.00 C                   ITER
```

```
0019.00 C                   ENDIF
0020.00  *
0021.00 C                   SELECT
0022.00 C                   WHEN      SOP = 2
0023.00 C                   EXSR      UPDATESR
0024.00 C                   WHEN      SOP = 4
0025.00 C                   EXSR      DELETESR
0026.00 C                   WHEN      SOP = 5
0027.00 C                   EXSR      DISPLAYSR
0028.00 C                   OTHER
0028.01 C                   EVAL      *IN87 = *ON
0029.00 C                   EVAL      MENUMSG = 'INVALID OPTION'
0030.00 C                   ENDSL
0031.00  *
0032.00 C                   ENDDO
0033.00  *
0034.00 C                   EVAL      *INLR = '1'
0035.00  *----------------------------------------------------------------
0036.00  *BOOK ENTRY SUBROUTINE
0037.00  *----------------------------------------------------------------
```

```
0038.00 C     ENTRYSR       BEGSR
0039.00 C                  DOW       *IN12 = '0'
0040.00 C                  EXFMT     BOOKENTRY
0040.01 C                  EVAL      ENTRYMSG = ' '
0041.00  *
0042.00 C                  IF        *IN12 = '1'
0043.00 C                  LEAVE
0044.00 C                  ENDIF
0045.00  *
0046.00 C                  EVAL      BOOKNO = EBOOKNO
0046.01 C                  EVAL      BOOKNAME = EBOOKNAME
0047.00 C                  EVAL      BOOKPRICE = EBOOKPRICE
0048.00 C                  WRITE     BOOKREC
0049.00 C                  EVAL      ENTRYMSG = 'Record added'
0050.00  *
0051.00 C                  ENDDO
0052.00 C                  EVAL      *IN12 = '0'
0053.00  *
0054.00 C                  ENDSR
0054.01  *----------------------------------------------------------------
```

```
0054.02  *BOOK UPDATE SUBROUTINE
0054.03  *----------------------------------------------------------------
0054.04 C     UPDATESR     BEGSR
0054.05 C                  DOW       *IN12 = '0'
0054.06 C                  EXFMT     BOOKUPDATE
0054.07 C                  EVAL      UPDATEMSG = ' '
0054.08  *
0054.09 C                  IF        *IN12 = '1'
0054.10 C                  LEAVE
0054.11 C                  ENDIF
0054.12  *
0054.13 C     UBOOKNO      CHAIN     BOOKFILE
0054.14 C                  IF        %FOUND()
0054.15 C                  EVAL      BOOKNAME = UBOOKNAME
0054.16 C                  EVAL      BOOKPRICE = UBOOKPRICE
0054.17 C                  UPDATE    BOOKREC
0054.18 C                  EVAL      UPDATEMSG = 'Record updated'
0054.19 C                  ELSE
0054.20 C                  EVAL      UPDATEMSG = 'Record not found'
0054.21 C                  ENDIF
```

```
0054.22  *
0054.23 C                   ENDDO
0054.24 C                   EVAL      *IN12 = '0'
0054.25  *
0054.26 C                   ENDSR
0054.27  *------------------------------------------------------------
0054.28  *BOOK DELETE SUBROUTINE
0054.29  *------------------------------------------------------------
0054.30 C     DELETESR      BEGSR
0054.31 C                   DOW       *IN12 = '0'
0054.32 C                   EXFMT     BOOKDELETE
0054.33 C                   EVAL      DELETEMSG = ' '
0054.34  *
0054.35 C                   IF        *IN12 = '1'
0054.36 C                   LEAVE
0054.37 C                   ENDIF
0054.38  *
0054.39 C     DBOOKNO       CHAIN     BOOKFILE
0054.40 C                   IF        %FOUND()
0054.42 C                   DELETE    BOOKREC
```

```
0054.43 C                   EVAL      DELETEMSG = 'Record deleted'
0054.44 C                   ELSE
0054.45 C                   EVAL      DELETEMSG = 'Record not found'
0054.46 C                   ENDIF
0054.47  *
0054.48 C                   ENDDO
0054.49 C                   EVAL      *IN12 = '0'
0054.50  *
0054.51 C                   ENDSR
```

```
0054.53  *BOOK DSPLY SUBROUTINE
0054.54  *------------------------------------------------------------
0054.55 C     DISPLAYSR     BEGSR
0054.56 C                   DOW       *IN12 = '0'
0054.57 C                   EXFMT     BOOKDSPLY
0054.58 C                   EVAL      DSMSG = ' '
0054.59  *
0054.60 C                   IF        *IN12 = '1'
0054.61 C                   LEAVE
0054.62 C                   ENDIF
0054.63  *
0054.64 C     DSBOOKNO      CHAIN     BOOKFILE
0054.65 C                   IF        %FOUND()
0054.66 C                   EVAL      DSBOOKNO = BOOKNO
0054.67 C                   EVAL      DSBOOKNAME = BOOKNAME
0054.68 C                   EVAL      DSBOOKPRIC = BOOKPRICE
0054.69 C                   ELSE
0054.71 C                   EVAL      DSMSG = 'Record not found'
0054.72 C                   ENDIF
```

```
0054.73  *
0054.74 C                   ENDDO
0054.75 C                   EVAL      *IN12 = '0'
0054.76  *
0054.77 C                   ENDSR
0055.00  *
```

Array programs

Types of Array

1) Compile time array : The compile time array means the elements of the array will be loaded before the execution of the programs i.e. at compile time. (static values)
We must declare in keyword command DIM (), CTDATA (), and PERRCD ()

15. Compile time array example

```
0001.00      Darray1           S              10    DIM(10)CTDATA
0001.01      Darray2           S              10    DIM(10)CTDATA
0001.02      Darray3           S               2    DIM(10)CTDATA PERRCD(2)
0001.03      DI                S               2S 0
0001.04       *
0001.05      C     array3(1)     DSPLY
0001.06      C     array3(2)     DSPLY
0001.07      C     array3(3)     DSPLY
0001.08      C     array3(4)     DSPLY
0001.09      C                   EVAL      *INLR = '1'
0001.10       *
0001.11 **
0001.12 value1
0001.13 value2
0001.14 value3
0001.15 value4
0001.16 value5
0001.17 value6
0001.18 value7
```

2) Pre-runtime array

In pre-runtime array, we maintain the array element in separate file. Hence, if we are making any change in array element we can just change this file containing the array element; we don't need to compile the source program again and again as in compile time array.

```
0002.00 DPRE_ARR          S              5    DIM(5) FROMFILE(FLAT01) PERRCD(1)
```

Flat files : Files without any structure

CRTPF COMMAND WITH RECORD length will create a flat file
CRTPF FILE(FLATFILE) RCDLEN(500)
To view flatfiles : DSPPFM FILE(EASYCLASS1/FLATFILE1)

```
     ********** Beginning of data **********
0001.00 FFLATFILE3 IT   F  500         DISK
0002.00 Darr1               S             25     DIM(15)FROMFILE(FLATFILE3)
0003.00 C      arr1(1)        DSPLY
0004.00 C      arr1(2)        DSPLY
0005.00 C      arr1(3)        DSPLY
0006.00 C      arr1(4)        DSPLY
0007.00 C                     EVAL       *INLR = '1'
     ********** End of data **********
```

3) Run time array

The run time array means the value will be loaded during the runtime only. (Dynamic)

16. Runtime array example

```
Darray1           S              10    DIM(10)
Darray2           S              10  0 DIM(10)
DI                S               2S 0
Dsum              S               3S 0
C                   EVAL      array1(1) = 'b'
C                   EVAL      array1(2) = 'c'
C                   EVAL      array1(3) = 'd'
C                   EVAL      array1(4) = 'a'
C                   EVAL      array1(5) = 'e'
C                   EVAL      array1(6) = 'f'
C                   EVAL      array1(7) = 'j'
C                   EVAL      array1(8) = 'h'
C                   EVAL      array1(9) = 'i'
C                   EVAL      array1(10) = 'g'
 *
C                   EVAL      array2(1) = 1
C                   EVAL      array2(2) = 2
C                   EVAL      array2(3) = 3
C                   EVAL      array2(4) = 4
C                   EVAL      array2(5) = 5
C                   EVAL      array2(6) = 6
C                   EVAL      array2(7) = 7
C                   EVAL      array2(8) = 8
C                   EVAL      array2(9) = 9
C                   EVAL      array2(10) = 10
 *
 *To display elements
C     'Initial arr:'DSPLY
C                   FOR       I = 1 TO 10 BY 1
C     array1(I)     DSPLY
C                   ENDFOR
 *
 * To search elements
C     'Searching:'  DSPLY
C                   EVAL      I = %LOOKUP('g':array1)
C     I             DSPLY
 *
 * To sort elements
C                   SORTA     array1
```

```
 *
C      'After sort :'DSPLY
C                    FOR       I = 1 TO 10 BY 1
C      array1(I)     DSPLY
C                    ENDFOR
 *
 * Sum of elements
C                    EVAL      sum = %XFOOT(array2)
C      sum           DSPLY
C                    EVAL      *INLR = '1'
```

DATA STRUCTURES

Data Structure is used-

1. To break fields into subfields
2. To Group fields
3. To change the format of the field
4. To Group non-contiguous data into contiguous format
5.To convert data.

Types of data structures in as/400:

I. program described data structure
II. EXTERNALLY DESCRIBED DATASTRUCTURE
III. MULTIPLE OCCURENCE DATASTRUCTURE
IV. INDICATOR DATA STRUCTURE : The indicator data structure is used to rename the indicators used in our program with the name that is more meaningful and understanding.
V. DATA AREA DATA STRUCTURE (SPECIFIED IN 'U')
VI. PROGRAMME STATUS DATASTRUCTURE (SPECIFIED IN 'S')

A program status data structure (PSDS) can be defined to make program exception/error information available to the program so that the necessary action can be taken for the unhandled exception. The exception /errors can be Divide by zero, array index out-of-bound, Invalid Date, Time or Timestamp value. The PSDS must be defined in the main source section; therefore, there is only one PSDS per module.

VII. FILE INFORMATION DATASTRUCTURE[minimum RRN/first RRN]

A file information data structure (INFDS) can be defined for each file to make file exception/error and file feedback information available to the program.

17. Simple data str (program described)

```
DDS1              DS
Dsub1                          10
Dsub2                          10S 0
Dsub3                          10
 *
C                   EVAL       sub1 = 'FLD1'
C                   EVAL       sub2 = 50
C                   EVAL       sub3 = 'FLD3'
 *
C     DS1           DSPLY
C                   EVAL       DS1 = 'AAAAAAAAAAGYWHDFJWEOIGFYWGF'
C     sub1          DSPLY
C     sub2          DSPLY
C     sub3          DSPLY
C                   EVAL       *INLR = '1'
```

18. Multioccurance data structure

```
0001.00 DDS1              DS                  occurs(3)
0002.00 DSUB1                          10
0003.00 DSUB2                          10S 0
0004.00 DSUB3                          10
0004.01 DI                S             1S 0
0005.00  *
0005.01 C                   EVAL       %occur(ds1) = 1
0006.00 C                   EVAL       SUB1 = 'FLD1'
0006.01 C                   EVAL       SUB2 = 50
0006.02 C                   EVAL       SUB3 = 'FLD2'
0006.03  *
0006.04 C                   EVAL       %occur(ds1) = 2
0006.05 C                   EVAL       SUB1 = 'FLD3'
0006.06 C                   EVAL       SUB2 = 51
0006.07 C                   EVAL       SUB3 = 'FLD5'
0006.08  *
0006.09 C                   EVAL       %occur(DS1) = 3
0006.10 C                   EVAL       SUB1 = 'FLD6'
0006.11 C                   EVAL       SUB2 = 52
```

```
0006.12 C                   EVAL       SUB3 = 'FLD8'
0018.01  *
0018.02 C                   EVAL       %occur(ds1) = 1
0018.03 C     DS1           DSPLY
0022.03  *
0022.04 C                   EVAL       %occur(ds1) = 2
0022.05 C     SUB2          DSPLY
0022.06 C     SUB3          DSPLY
0022.07  *
0023.00 C                   EVAL        *INLR = '1'
        ****************** End of data ****************
```

19. Externally described ds + array ds

Here, using a given customer no. and array %lookup() customer name is found and printed.

```
0000.01 FCUSTPF    IF   E             DISK
0001.00 DDS1            E DS                    EXTNAME('CUSTPF')DIM(2)QUALIFIED
0001.02 DI                S                2  0
0001.03 DJ                S                1  0
0001.04 DCUSTNO           S                3  0
0001.05 DCUSTNAME         S               10
0001.06 DCUSTOMER         S               10
0005.00  *
0005.01 C                   EVAL      I = 0
0005.02 C                   READ      CUSTREC
0005.03 C                   DOW       NOT %EOF()
0005.04 C                   EVAL      I=I+1
0005.05 C                   EVAL      DS1(I).CUSTNO = CUSTNO
0005.06 C                   EVAL      DS1(I).CUSTNAME = CUSTNAME
0005.07 C                   READ      CUSTREC
0005.08 C                   ENDDO
0005.09  *
0005.10 C                   EVAL      J=%LOOKUP(101:DS1(*).CUSTNO)
0005.11 C                   IF        J>0
0005.12 C                     EVAL      CUSTOMER = DS1(J).CUSTNAME
0005.13 C                     ENDIF
0005.14  *
0005.16 C     CUSTOMER        DSPLY
0023.00 C                     EVAL      *INLR = '1'
        ****************** End of data ********************************
```

20. Program status ds(PSDS)

```
0001.00 DMYPSDS          SDS
0002.00 D PROC_NAME         *PROC
0003.00 D PGM_STATUS        *STATUS
0004.00 D PARMS             *PARMS
0004.01 D PGM_NAME                1     10
0004.02 D LINE_NUM               21     28
0004.03 D DATE                  191    198
0005.00 D JOB_NAME              244    253
0006.00 D USER                  254    263
0007.00 D SRC_FILE              304    313
0008.00 D SRC_LIB               314    323
0008.01  *
0008.02 C     PROC_NAME     DSPLY
0008.03 C     PGM_STATUS    DSPLY
0008.04 C     PARMS         DSPLY
0008.05 C     PGM_NAME      DSPLY
0008.06 C     LINE_NUM      DSPLY
0008.07 C     DATE          DSPLY
0008.08 C     JOB_NAME      DSPLY
```

```
0008.09 C     SRC_FILE      DSPLY
0008.10 C     USER          DSPLY
0008.11 C     SRC_LIB       DSPLY
0009.00 C                   EVAL      *INLR = '1'
        ****************** End of data ************
```

21. INFDS

```
0001.00 HOPTION(*NODEBUGIO:*SRCSTMT)
0002.00 FORDERPF1  UF A E           K DISK
0003.00 FSFLPGM2   CF   E             WORKSTN SFILE(ORDERSFL:RRN)INFDS(DS1)
0004.00 DRRN              S              4  0
0005.00 DDS1              DS
0006.00 DSUB1                   378    379I 0
0007.00  *
0008.00 C                   EVAL      RCD_NBR = 1
0009.00 C                   DOW       *IN03 = '0'
0010.00  *
0011.00 C                   EXSR      CLRSFL
0011.01 C                   EXSR      LOADSFL
0011.02 C                   EXSR      DSPLYSFL
0012.00 C                   EXSR      PROCSFL
0013.00  *
0014.00 C                   ENDDO
0015.00 C                   EVAL      *INLR = '1'
0016.00  ******************************************************************************
0017.00 C     CLRSFL        BEGSR
0017.01 C                   EVAL      *IN93 = '1'
0017.02 C                   WRITE     ORDERCTL
0017.03 C                   EVAL      *IN93 = '0'
0017.04 C                   ENDSR
0018.00  ***********************************************
0019.00 C     LOADSFL       BEGSR
0019.01 C                   EVAL      RRN = 0
0019.02 C     *LOVAL        SETLL     ORDERPF1
0019.03 C                   READ      ORDERPF1
0019.04 C                   DOW       NOT %EOF()
0019.05 C                   EVAL      RRN = RRN + 1
0019.06 C                   WRITE     ORDERSFL
0019.07 C                   IF        SPOS = ORDERNO
0019.08 C                   EVAL      RCD_NBR = RRN
0019.09 C                   ENDIF
0019.10 C                   READ      ORDERPF1
0019.11 C                   ENDDO
0020.00 C                   ENDSR
0021.00  ***********************************************
0022.00 C     DSPLYSFL      BEGSR
```

```
0022.01 C                   WRITE     FOOTER
0022.02 C                   EVAL      *IN92 = '1'
0022.03 C                   IF        RRN > 0
0022.04 C                   EVAL      *IN91 = '1'
0022.05 C                   EVAL      *IN94 = '1'
0022.06 C                   ENDIF
0022.07 C                   EXFMT     ORDERCTL
0022.08 C                   IF        SUB1 > 0
0022.09 C                   EVAL      RCD_NBR = SUB1
0022.10 C                   ELSE
0022.11 C                   EVAL      RCD_NBR = 1
0022.12 C                   ENDIF
0022.13 C                   EVAL      *IN92 = '0'
0022.14 C                   EVAL      *IN91 = '0'
0022.15 C                   EVAL      *IN94 = '0'
0023.00 C                   ENDSR
0024.00  ********************************************************
0025.00 C     PROCSFL       BEGSR
0025.01 C                   IF        *IN03= '1'
0025.02 C                   LEAVESR
0025.03 C                   ENDIF
0025.04 C                   IF        RRN = 0
0025.05 C                   LEAVESR
0025.06 C                   ENDIF
0025.07 C
0026.00 C                   ENDSR
        ******************** End of data ********
```

22. Reading from itempf and writing into flatfile

```
0001.00 FITEMPF    IF   E           DISK
0002.00 FFLATFILE3 UF A E           DISK    RENAME(FLATFILE3:RECORD)PREFIX('P_')
0003.00 DP_FLATFILE3    E DS                EXTNAME('ITEMPF')
0004.00  *
0005.00 C                   READ      ITEMPF
0006.00 C                   DOW       NOT %EOF()
0007.00 C                   WRITE     RECORD
0008.00 C                   READ      ITEMPF
0009.00 C                   ENDDO
0010.00 C                   EVAL      *INLR = '1'
       ******************** End of data ********************************************
```

23. Block error handling

```
0001.00 Dnum1              S               2  0
0002.00 Dnum2              S               2  0
0003.00 Dquotient          S               2  0
0003.01 Dmsg               S              20
0004.00  *
0004.01 C                   EVAL       num1 = 10
0004.02 C                   EVAL       num2 = 0
0004.03  * Using MONITOR keyword to handle division by zero error
0005.00 C                   MONITOR
0006.00 C                   EVAL       quotient = num1/num2
0007.00 C     quotient      DSPLY
0008.00 C                   ON-ERROR
0009.00 C                   EVAL       msg = 'Division by zero error'
0010.00 C     msg           DSPLY
0011.00 C                   ENDMON
0012.00  *
0013.00 C                   EVAL       *INLR = '1'
        ****************** End of data ********************************
```

24. File or single opcode error handling – write,read,chain,setll

```
0001.00 FEMPPF     UF A E           K DISK
0001.01 Dnum               S             10  0
0002.00 Dmsg               S             20
0003.00  *
0004.01 C                   EVAL       EMPNO = 101
0004.02 C                   EVAL       EMPNAME = 'RAVI'
0004.03 C                   WRITE(E)   EMPREC
0007.00 C                   IF         %ERROR()
0008.00 C                   EVAL       msg = 'Duplicate record'
0008.01 C     msg           DSPLY
0009.00 C                   ENDIF
0011.00 C                   EVAL       *INLR = '1'
        ****************** End of data ********************************
```

25. Take accno as character input from user and display its transactions

```
Columns . . . :    6  80                          Browse                        GOPZLIB/QRPGLESRC
SEU==> _____________________________________________________________________________ ACCPRGM
FMT FX FFilename++IPEASF.....L.....A.Device+.Keywords++++++++++++++++++++++++++++++
       ************** Beginning of data ********************************************************************************
0001.00 FTRANLF    IF   E           K DISK                                    250109
0002.00 Dmsg             S              30                                    250109
0002.01 Dacc             S              14                                    250109
0002.02 Daccount         S              14  0                                 250109
0003.00  *                                                                    250109
0004.00 C     *ENTRY       PLIST                                             250109
0005.00 C                  PARM                    acc                       250109
0006.00  *                                                                    250109
0006.01 C                  EVAL      account = %dec(acc:14:0)                250109
0007.00 C     account      SETLL     TRANLF                                  250109
0008.00 C     account      READE(E)  TRANLF                                  250109
0008.01  *                                                                    250109
0008.02 C                  IF        %ERROR()                                250109
0008.03 C     'Cant read'  DSPLY                                             250109
0008.04 C                  ENDIF                                             250109
0008.05  *                                                                    250109
0009.00 C                  DOW       NOT %EOF()                              250109
0010.00 C                  EVAL      msg = %CHAR(TRANNO)+' '                 250109
0011.00 C                            +%CHAR(TRANAMOUNT)+' '+%TRIM(TRANTYPE)  250109

F3=Exit    F5=Refresh    F9=Retrieve    F10=Cursor    F11=Toggle    F12=Cancel
F16=Repeat find          F24=More keys
```

```
0012.00 C      msg             DSPLY
0013.00 C      account         READE      TRANLF
0014.00 C                      ENDDO
0015.00  *
0016.00 C                      EVAL       *INLR = '1'
0017.00 C*                     RETURN
        ********************* End of data *********************
```

26. Copybook

```
Columns . . . :    6  80                          Browse                        GOPZLIB/QRPGLESRC
SEU==> _____________________________________________________________________________ COPYBOOK
FMT C  CL0N01Factor1+++++++Opcode&ExtFactor2+++++++Result++++++++Len++D+HiLoEq....
       ************** Beginning of data ********************************************************************************
0001.00 c      'END OF PRGM' DSPLY                                        250109
       ****************** End of data ***********************************************************************************
```

```
Columns . . . :    6  80                          Browse                        GOPZLIB/QRPGLESRC
SEU==> _____________________________________________________________________________ PROG5
FMT D  DName++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++++++
       ************** Beginning of data ********************************************************************************
0001.00 DFNAME            S             10                                  241227
0002.00 DLNAME            S             10                                  241227
0003.00 DFULLNAME         S             20                                  241227
0004.00 C      *entry       plist                                           241227
0005.00 C                   parm                    FNAME                   241227
0006.00 C                   parm                    LNAME                   241227
0007.00 C                   EVAL      FULLNAME = %TRIM(FNAME)+' '+%TRIM(LNAME)241227
0008.00 C      FULLNAME     DSPLY                                           241227
0008.01  /COPY COPYBOOK                                                     250109
0009.00 C                   EVAL      *INLR = '1'                           241227
       ****************** End of data ***********************************************************************************
```

27. Freeformat date

```
Columns . . . :    6  80                          Browse                    GOP2LIB/QRPGLESRC
SEU==>                                                                        DATEFREEFM
FMT H  HKeywords+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       ********** Beginning of data **************************************************************************************
0001.00 HDATFMT(*ISO)                                                      250113
0002.00 DDATE1           S               D                                 250113
0002.01 DDATE2           S               D                                 250113
0002.02 DDATE3           S               D                                 250113
0002.03 DDATE4           S               D                                 250113
0003.00 DCHAR1           S              10    INZ('2024-01-15')            250113
0004.00 DCHAR2           S               8    INZ('20240115')              250113
0005.00 DCHAR3           S               6    INZ('011325')                250113
0005.01  //Conversion of char2 to date                                      250113
0006.00      DATE1 = %DATE(CHAR2:*ISO0);                                   250113
0007.00      DSPLY CHAR2;                                                  250113
0008.00      DSPLY DATE1;                                                  250113
0009.00  //Conversion of char3 to date                                      250113
0010.00      DATE1 = %DATE(CHAR3:*MDY0);                                   250113
0011.00      DSPLY CHAR3;                                                  250113
0012.00      DSPLY DATE1;                                                  250113
0012.01  //Adding days to date                                            250113
0012.02      DATE2 = DATE1 + %DAYS(20);                                    250113
0012.03      DSPLY DATE2;                                                  250113

F3=Exit    F5=Refresh    F9=Retrieve    F10=Cursor    F11=Toggle    F12=Cancel
F16=Repeat find          F24=More keys
                                    (C) COPYRIGHT IBM CORP. 1981, 2013.
```

```
0012.04  //Adding months to date
0012.05       DATE3 = DATE1 + %MONTHS(4);
0012.06       DSPLY DATE3;
0012.07  //Adding year to date
0012.08       DATE4 = DATE1 + %YEARS(2);
0012.09       DSPLY DATE4;
0013.00       *INLR = '1';
          ********** End of data *********
```