

RPGLE ASSIGNMENT (11.1.2025)

1. Create rpgle program to take account no as input and display sum of transactions. There should be 2 displays one for credit and other for debit. We can use transaction pf for this.

```
0001.00                                     UNIQUE
0002.00          R TRANREC
0003.00          TRANNO          16  0
0004.00          TRANAMOUNT      10  2
0005.00          TRANTYPE        10
0006.00          ACCNO           14  0
0006.01          TRANDATE         L
0006.02          ACCSTATUS        1
0006.03          ACCENDDATE       L
0007.00          K TRANNO
***** End of data *****
```

```
Columns . . . : 1 80      Browse      GOP2LIB/QDDSSRC
SEU==>      TRANLF
FMT LF .....A.....T.Name+++++.Len++TdP.....Functions+++++
***** Beginning of data *****
0001.00          R TRANREC          PFILE(TRANPF)          241224
0002.00          K ACCNO           241227
***** End of data *****
```

- a) H SPEC (Control specifications, Compiler options) [Keyword field only]
OPTION(*{NO}GEN *{NO}EXT *{NO}SRCSTMT) *{NO}DEBUGIO)
***{NO}GEN:** Determines whether or not to generate a compiled object.
***{NO}SRCSTMT:** If *NOSRCSTMT is coded or this statement is excluded (*NOSRCSTMT is the default), statements will be re-numbered when compiling the program. If your program gets a run-time error, the statement indicated will be virtually useless unless you have a copy of compile. *SRCSTMT will prevent the compiler from renumbering the program's statements.
***{NO}DEBUGIO:** If you use the interactive source debugger to step through a program, you'll notice that the debugger will break many times on every I/O statement. This is because a separate breakpoint is inserted for every field returned from the I/O buffer. This is not a major problem, and very inconvenient. Coding *NODEBUGIO will break only once for each I/O statement.
DATFMT (FMT)
Specify this keyword to define the default format for date fields within the program. Choose one of the following date formats:
 - *MDY (mm/dd/yy)
 - *DMY (dd/mm/yy)
 - *YMD (yy/mm/dd)
 - *ISO (yyyy-mm-dd)
 - *USA (mm/dd/yyyy)
 - *EUR (dd.mm.yyyy)e.g. H datfmt(*MDY)

By default FMT will be *ISO.

b) F SPEC (File description)

```

Prompt type . . .      F      Sequence number . . .  ' ' ' ' ' '

File      File      End of      File
Filename  Type      Designation  File      Addition      Sequence
-----
File      Record    Limits      Length of    Record
Format    Length    Processing  Key Field    Address Type
-----
File
Organization      Device      Keywords
-----
Comment
-----

```

File name: Mention here name of the file that you are going to use in your program.

File Type: Mention the file type as I,O,U,C where I= INPUT, O=OUTPUT,U=UPDATE,C=COMBINED.

File Designation: Mention the File Designation as P,S,F where P=PRIMARY,S=SECONDARY,F=FULL PROCEDURAL.

Primary File = Record will be processed in the order; from start to end by rpg program cycle i.e. OPEN, READ, PROCESS, CLOSE. User can't change this order. There can be only one primary file in the program.

Secondary Files = Secondary files apply to programs that do multifile processing. All of the files involved in multifile processing, except the primary file, are secondary files.

Full Procedural file = User can control any order by rpg program opcode. User can change any order by rpg opcode. With full procedural files the programmer determines which record or a block of records to be read by way of the value of the key field used.

File Addition: Mention 'A' if you want to add record to the DISK FILE. In update mode of file, use 'A' file designation.

File Format: Mention if the file is program described or externally described.

Record Address Type: Mention this field entry as 'K' if the file is a keyed file, blank if the file is a sequential access file or based on RRN.

Device: Mention the device as DISK, PRINTER or WORKSTN.

Keywords : PREFIX(prefix{:nbr_of_char_replaced}), RENAME(Ext_format:Int_format), SFILE(recformat:rrnfield), INFDS(data structure name),

INFSR(subroutine_name), OFLIND(indicator)

*PSSR - PROGRAM STATUS SUBROUTINE

WHEN PROGRAM ENDS ABNORMALLY, CONTROL WILL GO TO PSSR AND ERROR CONDITIONS WILL BE HANDLED THERE. IT CAN BE HANDLED ONLY BY ENDSR. IT CAN BE USED FOR LOGGING ERRORS.

```
C *PSSR      BEGSR
C 'ERROR'    DSPLY
C              ENDSR  '*CANCL'
```

*CANCL WILL END THE PROGRAM.
 *RETRY WILL RETRY TO RUN Again
 *GETIN

INFSR - FOR FILE LEVEL. INFSR IS A KEYWORD NOT A SUBROUTINE UNLIKE *PSSR.

```
FFILENAME IF      E      A DISK      INFSR(SR)
```

SR SHOULD GET EXECUTED WHEN THERE IS FILE ISSUE.
 SR CAN BE REPLACED BY *PSSR TOO.

c) D SPEC(Definition specifications)

Prompt type . . .	D	Sequence number . . . '			
-------------------	---	---	--	--	--

Keywords : CONST(value), LIKE(RPG_name), OCCURS, CTDATA, DIM, PERRCD(numeric_constant), INZ

d) C SPEC (Calculation specifications)

Prompt type . . .		C	Sequence number . . .				''''''
Level	N01	Factor 1	Operation			Factor 2	Result
—	—	—	—			—	—
		Decimal					
Length	Positions	HI	LO	EQ	Comment		
—	—	—	—	—	—		

N01 (Conditioning Indicator) :

Factor 1 : represents the name or literal on which the actual operation is to be performed.

Operation : EVAL, DOW, DOU, FOR, IF, SORTA, READ, READC, READP.....

Factor 2 :

Result :

Field length :

Decimal positions :

e) **P SPEC**(Procedure specifications) :

Deals with procedures of different languages (RPG ILE).

f) **O SPEC** (Output specifications) :

For output settings

```
0001.00 FTRANLF      IF      E              K DISK
0002.00 Dcreditsum          S              10  2
0003.00 Ddebitsum          S              10  2
0003.01 Daccount           S              14  0
0003.02 Dmsg               S              50
0004.00 *
0004.01 C      *ENTRY      PLIST
0004.02 C              PARM              account
0004.03 C              EVAL      creditsum = 0
0004.04 C              EVAL      debitsum = 0
0005.00 C      account     SETLL      TRANLF
0006.00 C      account     READE      TRANLF
0007.00 C              DOW      NOT %EOF()
0007.01 *
0008.00 C              IF      TRANTYPE = 'CREDIT'
0009.00 C              EVAL      creditsum = creditsum + TRANAMOUNT
0010.00 C              ELSEIF   TRANTYPE = 'DEBIT'
0011.00 C              EVAL      debitsum = debitsum + TRANAMOUNT
0012.00 C              ENDIF
```

```
0013.00 *
0013.01 C      account     READE      TRANLF
0014.00 C              ENDDO
0015.00 *
0016.00 C              EVAL      msg = %trim(%char(account))+ ' '
0017.00 C              + 'Credit_sum : '+%trim(%char(creditsum))
0018.00 C              + ' Debit_sum : '+%trim(%char(debitsum))
0019.00 C      msg         DSPLY
0020.00 C              EVAL      *INLR = '1'
***** End of data *****
```

RETURN VS INLR

INLR => ALL THE FILES WILL BE CLOSED, CLEAR THE MEMORY.

RETURN => CLOSING WILL NOT HAPPEN, VARIABLE VALUES WILL NOT BE CLEARED
UNLIKE INLR. IT WILL CONFLICT WITH OTHER PROGRAM LIKE IF NEXT PROGRAM DONT

HAVE INITIALIZATION FOR THOSE VARIABLES USING AGAIN, IT WILL RETAIN THE VALUE OF PREV PROGRAM.

ADVANTAGE OF RETURN

```
DSPLY 103 Credit_sum : 18900.00 Debit_sum : 100.00
```

2. Create rpgle program to populate run time array from all unique account no in transaction file.

```
0001.00 FTRANLF      IF      E              K DISK
0002.00 Darr          S              14  0 dim(10)
0002.01 DI            S              2  0
0002.02 Dprevacc      S              14  0
0002.03 Dcuracc       S              14  0
0003.00 *
0004.00 C      *LOVAL      SETLL      TRANLF
0005.00 C              READ      TRANLF
0006.00 C              EVAL      prevacc = ACCNO
0007.00 C              EVAL      I = 1
0008.00 *
0009.00 C              DOW      NOT %EOF(TRANLF)
0010.00 C              READ      TRANLF
0011.00 C              EVAL      curacc = ACCNO
0012.00 C              IF      prevacc <> curacc
0013.00 C              EVAL      arr(I) = prevacc
0014.00 C              EVAL      I = I + 1
0014.02 C              EVAL      prevacc = curacc
0015.00 C              ENDIF
0016.00 C              ENDDO
0016.01 C              EVAL      arr(I) = curacc
0017.00 *
0017.01 C              FOR      I = 1 TO 9 BY 1
0017.02 C              IF      arr(i) <> 0
0018.00 C      arr(I)      DSPLY
0018.01 C              ENDIF
0018.02 C              ENDFOR
0019.00 C              EVAL      *INLR = '1'
***** End of data *****
```

```
DSPLY 102
DSPLY 103
DSPLY 104
```

3. Create a rpgle program to take student id as input and display student name on screen
use studpf for this .

```

Columns . . . : 1 80 Browse GOP2LIB/QDDSSRC
SEU==>
FMT PF .....A.....T.Name+++++RLen++TDpB.....Functions+++++STUDPF
***** Beginning of data *****
0002.00 R STUDREC 250108
0003.00 STUDID 3S 0 250108
0004.00 STUDNAME 10 250108
0005.00 SUB 10 250108
0007.00 MARKS_OBT 5S 2 250108
0008.00 TOT_MARKS 5S 2 DFT(300.00) 250108
0009.00 R STUDID 250108
***** End of data *****

```

```

0001.00 FSTUDPF IF E K DISK
0002.00 FSTUDSCN2 CF E WORKSTN
0003.00 *
0004.00 C DOW *IN03 = '0'
0005.00 C EXFMT STUDDSPY
0005.01 C EVAL SMSG = ' '
0005.02 C EVAL SSTUDNAME = ' '
0006.00 C IF *IN03 = '1'
0007.00 C LEAVE
0008.00 C ENDIF
0009.00 C SSTUDID CHAIN STUDPF
0009.01 C IF %FOUND()
0010.00 C EVAL SSTUDNAME= STUDNAME
0010.01 C ELSE
0010.02 C EVAL SMSG = 'Record not found'
0011.00 C ENDIF
0014.00 C ENDDO
0015.00 *
0016.00 C EVAL *INLR = '1'

```

ENTER STUDENT ID :

STUDENT NAME : ANN

F3=EXIT

```

ENTER STUDENT ID : 140

STUDENT NAME :

█

F3=EXIT

```

```

ENTER STUDENT ID : 

```

```

STUDENT NAME :

```

```

Record not found

```

```

F3=EXIT

```

```

ENTER STUDENT ID : 

```

```

STUDENT NAME :

```

```

Record not found

```

```

F3=EXIT

```

```

ENTER STUDENT ID : 

```

```

STUDENT NAME :

```

```

Record not found

```

```

F3=EXIT

```

```

ENTER STUDENT ID : 

```

```

STUDENT NAME :

```

```

Record not found

```

```

F3=EXIT

```

4. Add a new field as account status(char 1) in transaction pf.

Create a screen to take account no as input and mark all records in transaction pf with satus 'I' for that account .

screen name can be : account inactivate screen.

```

0001.00                                     UNIQUE
0002.00          R  TRANREC
0003.00          TRANNO          16  0
0004.00          TRANAMOUNT      10  2
0005.00          TRANTYPE        10
0006.00          ACCNO          14  0
0006.01          TRANDATE          L
0006.02          ACCSTATUS        1
0006.03          ACCENDDATE        L
0007.00          R  TRANNO
          ***** End of data *****

```

```
Columns . . . : 1 80 Browse GOPZLIB/QDDSSRC
SEU==>
FMT LF .....A.....T.Name+++++.Lent+TDpB.....Functions+++++
***** Beginning of data *****
0001.00 R TRANREC PFILE (TRANPF) 241224
0002.00 K ACCNO 241227
***** End of data *****
```

```

0001.00 FTRANLF      UF A E          K DISK
0001.01 FACCSTSSCN CF  E          WORKSTN
0003.00  *
0006.01 C           DOW      *IN03 = '0'
0006.02 C           EXFMT    ACCNTINPUT
0006.03 C           EVAL     SMSG = ' '
0006.04  *
0006.05 C           IF       *IN03 = '1'
0006.06 C           LEAVE
0006.07 C           ENDIF
0006.08  *
0007.00 C      SACCNO      SETLL    TRANLF
0008.00 C      SACCNO      READE     TRANLF
0008.01  *
0009.00 C           DOW      NOT %EOF()
0010.00 C           EVAL     ACCSTATUS = 'I'
0011.00 C           UPDATE   TRANREC
0012.00 C           EVAL     SMSG = 'ACCOUNT DEACTIVATED'
0013.00 C      SACCNO      READE     TRANLF

```

```

0014.00 C           ENDDO
0015.00  *
0015.01 C           ENDDO
0016.00 C           EVAL     *INLR = '1'

```

ACCOUNT INACTIVE SCREEN

ENTER THE ACCOUNT NUMBER :

F3=EXIT

5. Create a rpgle program to take account as input and display all transaction of that account on screen.

In footer it should show sum.


```

0001.00 HOPTION(*NODEBUGIO:*SRCSTMT)
0002.00 FTRANLF      UF A E          K DISK
0003.00 FACCSUBFILECF E          WORKSTN SFILE(ACCSFL:RRN)
0004.00 *
0005.00 DRRN          S              4 0
0005.01 Dtotal        S              10 2
0006.00 *
0006.02 C              DOW          *IN03 = '0'
0007.00 C              EXSR          CLRSFL
0008.00 C              EXSR          LOADSFL
0009.00 C              EXSR          DSPSFL
0009.01 C              EXSR          PROCSFL
0009.02 *
0011.00 C              ENDDO
0012.00 *
0013.00 C              EVAL          *INLR = *ON
0014.00 *
0015.00 *-----
0016.00 *CLEAR SUBFILE

```

```

0017.00 *-----
0018.00 C      CLRSFL      BEGSR
0018.01 C              EVAL          *IN93 = '1'
0018.02 C              WRITE        ACCCTL
0018.04 C              EVAL          *IN93 = '0'
0019.00 C              ENDSR
0019.01 *-----
0019.02 *LOAD SUBFILE FROM DATABASE
0019.03 *-----
0019.04 C      LOADSFL      BEGSR
0019.05 C              EVAL          RRN = 0
0019.06 C              EVAL          SUM = 0
0019.07 C      SACCNO      SETLL      TRANLF
0019.08 C      SACCNO      READE      TRANLF
0019.09 C              EVAL          TOTAL = 0
0019.16 C              DOW          NOT %EOF(TRANLF)
0019.17 C              IF          TRANTYPE= 'CREDIT'
0019.19 C              EVAL          total = total + TRANAMOUNT
0019.20 C              ELSEIF      TRANTYPE= 'DEBIT'
0019.21 C              EVAL          total = total - TRANAMOUNT

```

```

0019.22 C          ENDIF
0019.23 C          EVAL      SUM = TOTAL
0019.24 C          EVAL      RRN = RRN + 1
0019.25 C          WRITE     ACCSFL
0019.27 *
0019.28 C          SACCNO     READE     TRANLF
0019.29 C          ENDDO
0019.31 C          ENDSR
0019.32 *-----
0019.33 *DISPLAY SUBFILE
0019.34 *-----
0019.35 C          DSPSFL      BEGSR
0019.36 C          WRITE      FOOTER
0019.37 C          EVAL      *IN92 = '1'
0019.38 C          IF        RRN > 0
0019.39 C          EVAL      *IN91 = '1'
0019.40 C          EVAL      *IN94 = '1'
0019.41 C          ENDIF
0019.42 C          EXFMT      ACCCTL
0019.43 C          EVAL      *IN91 = '0'

```

```

0019.44 C          EVAL      *IN92 = '0'
0019.45 C          EVAL      *IN94 = '0'
0019.46 C          ENDSR
0020.00 *
0020.01 *-----
0020.02 * PROCESS SUBROUTINE
0020.03 *-----
0020.04 C          PROCSFL      BEGSR
0020.05 C          IF          *IN03 = '1'
0020.06 C          LEAVESR
0020.07 C          ENDIF
0020.08 *
0020.09 C          IF          RRN = 0
0020.10 C          LEAVESR
0020.11 C          ENDIF
0020.16 *
0020.33 C          ENDSR
***** End of data *****

```

Work with Display Records						
File :		QDDSSRC		Member :		ACCSUBFILE
Library :		GOPZLIB		Source type :		DSPF
Type options, press Enter.						
1=Add		2=Edit comments		3=Copy		4=Delete
7=Rename		8=Select keywords		12=Design image		
Opt	Order	Record	Type	Related Subfile	Date	DDS Error
—	10	ACCSFL	SFL		01/11/25	
—	20	ACCCTL	SFLCTL	ACCSFL	01/11/25	
—	30	FOOTER	RECORD		01/11/25	

ACCOUNT DETAILS

ENTER THE ACCOUNT NUMBER :

TRANNO	TRANAMOUNT	TRANSTYPE	TRANDATE	ACCSTATUS
00000000000000102	0000550000	CREDIT	0001-01-01	I
00000000000000121	0000800000	CREDIT	0001-01-01	I
00000000000000122	0000500000	CREDIT	2025-01-10	I
00000000000000124	0000010000	DEBIT	2025-01-10	I
00000000000000125	0000040000	CREDIT	2025-01-10	I

Bottom

SUM : 18800.00

F3=EXIT

6. Create a rpgle program to take account no as input and create a report showing debit and credit transactions separately and sum of each type.

```
Columns . . . : 1 80 Browse GOPZLIB/QDDSSRC
SEU==> TRANLF1
FMT LF .....A.....T.Name+++++.Len+TDpB.....Functions+++++
***** Beginning of data *****
0001.00 R TRANREC PFILE(TRANPF) 241224
0002.00 K ACCNO 241227
0003.00 K TRANTYPE 250111
***** End of data *****
```

```
0001.00 FTRANLF1 IF E K DISK
0002.00 FACCREPORT O E PRINTER OFLIND(*IN99)
0002.01 Daccount S 14 0
0002.02 DTOT S 10 2
0002.03 *Dcreditsum S 10 2
0002.04 *Ddebitsum S 10 2
0003.00 *
0004.00 C *ENTRY PLIST
0005.00 C PARM account
0006.00 *
0007.00 C KLIST1 KLIST
0008.00 C KFLD ACCNO
0009.00 C KFLD TRANTYPE
0009.01 *
0009.02 C EVAL DATE = %DATE()
0009.03 C WRITE HEADER
0009.05 C EVAL ACC_NO = account
0009.06 C WRITE HEADING1
0009.07 C WRITE HEADING2
```

```

0010.00  *
0011.00 C          EVAL      TRANTYPE = 'CREDIT'
0011.01 C          EVAL      TRAN_TYPE = 'CREDIT'
0011.02 C          WRITE      HEADING3
0012.00 C          EVAL      ACCNO = account
0012.01 C      KLIST1  SETLL    TRANLF1
0013.00 C      KLIST1  READE     TRANLF1
0014.00 C          EVAL      TOTAL_AMT = ' '
0014.01 C          EVAL      TOT = 0
0014.02  *
0015.00 C          DOW        NOT %EOF(TRANLF1)
0015.01  *
0015.02 C          IF         *IN99 = '1'
0015.03 C          WRITE      HEADER
0015.04 C          EVAL      ACC_NO = account
0015.05 C          WRITE      HEADING1
0015.06 C          WRITE      HEADING2
0015.07 C          WRITE      HEADING3
0015.08 C          EVAL      *IN99 = '0'
0015.09 C          ENDIF

```

```

0015.10  *
0015.11 C          WRITE      RECORDS
0016.00 C          IF         TRANTYPE = 'CREDIT'
0017.00 C          EVAL      TOT = TOT + tranamount
0017.01 C          EVAL      TOTAL_AMT = %EDITC(TOT:'3')
0018.03 C          ENDIF
0018.04 C      KLIST1  READE     TRANLF1
0019.00 C          ENDDO
0020.00  *
0021.00 C          WRITE      SUMMARY
0021.01  *
0021.02 C          EVAL      TRANTYPE = 'DEBIT'
0021.03 C          EVAL      TRAN_TYPE = 'DEBIT'
0021.04 C          WRITE      HEADING3
0021.05 C          EVAL      ACCNO = account
0021.06 C      KLIST1  SETLL    TRANLF1
0021.07 C      KLIST1  READE     TRANLF1
0021.08 C          EVAL      TOTAL_AMT = ' '
0021.09 C          EVAL      TOT = 0
0021.10  *

```

```

0021.11 C          DOW          NOT %EOF(TRANLF1)
0021.12 *
0021.13 C          IF          *IN99 = '1'
0021.14 C          WRITE        HEADER
0021.15 C          EVAL          ACC_NO = account
0021.16 C          WRITE        HEADING1
0021.17 C          WRITE        HEADING2
0021.18 C          WRITE        HEADING3
0021.19 C          EVAL          *IN99 = '0'
0021.20 C          ENDIF
0021.21 *
0021.22 C          WRITE        RECORDS
0021.23 C          IF          TRANTYPE = 'DEBIT'
0021.24 C          EVAL          TOT = TOT + tranamount
0021.25 C          EVAL          TOTAL_AMT = %EDITC(TOT:'3')
0021.26 C          ENDIF
0021.27 C          KLIST1       READE        TRANLF1
0021.28 C          ENDDO
0021.29 *
0021.30 C          WRITE        SUMMARY

```

```

0022.00 C          EVAL          *INLR = '1'
          ***** End of data *****

```

```

Columns . . . :   1 121                Design Report                GOPZLIB/QDDSSRC
RLU==> |
BASE    ..... 1 ..... 2 ..... 3 ..... 4 ..... 5 ..... 6 ..... 7 ..... 8 ..... 9 ..... 0 ..... 1 ..... 2
----- Start of Page 001 -----
FLD1    <.....> <.....>                <.....>                <.....> <>
HEADER  DATE : XXXXXXXXXX                CREDIT AND DEBIT REPORT    PAGE : 99
FLD1    <.....> <.....>
HEADING  ACCOUNT_NO : 99999999999999
FLD1    <.....> <.....>
HEADING  TRANSACTIONS : XXXXXXXXXX
FLD1    <.....>                <.....>                <.....>
HEADING  TRANSACTION_NO      TRANSACTION_AMOUNT      TRANSACTION_DATE
FLD1    <.....>                <.....>                <.....>
RECORDS  9999999999999999      9999999999      XXXXXXXXXX
FLD1    <.....> <.....>
SUMMARY  SUM_OF_TRANSACTION_AMOUNT XXXXXXXXXX
----- End of Report -----

```

```

DATE : 2025-01-14                CREDIT AND DEBIT REPORT                PAGE : 01
ACCOUNT_NO : 00000000010300
TRANSACTIONS :
TRANSACTION_NO      TRANSACTION_AMOUNT      TRANSACTION_DATE
0000000000000102    0000550000      0001-01-01
0000000000000121    0000800000      0001-01-01
0000000000000122    0000500000      2025-01-10
0000000000000125    0000040000      2025-01-10
SUM_OF_TRANSACTION_AMOUNT      18900.0
TRANSACTION_NO      TRANSACTION_AMOUNT      TRANSACTION_DATE
0000000000000124    0000010000      2025-01-10
SUM_OF_TRANSACTION_AMOUNT      100.0

```

7. Create a new table employee salary extract.

This table should contain employee id , employee name , month , salary.

Create rpgle program to populate this file it should get details from employee pf and salary pf . It should populate details of all employees.

```

Columns . . . :   1  80                                Browse                                GOPZLIB/QDDSSRC
SEU==>
FMT PF .....A.....T.Name+++++RLen++TDpB.....Functions+++++
***** Beginning of data *****
0001.00          R EXTEMPREC                                250112
0002.00          EMPID              10  0                    250112
0003.00          ENAME              20                      250112
0004.00          MONTH              10                      250112
0005.00          SALARY              10  2                    250112
0006.00          K EMPID                                250112
***** End of data *****

```

```

Columns . . . :   1  80                                Browse                                GOPZLIB/QDDSSRC
SEU==>
FMT PF .....A.....T.Name+++++RLen++TDpB.....Functions+++++
***** Beginning of data *****
0002.00          R SALREC                                241225
0003.00          SALNO              10  0                    241225
0004.00          EMPNO              10  0                    241225
0005.00          SALAMT              10  2                    241225
0005.01          SMONTH              10                      250112
0006.00          K EMPNO                                250113
***** End of data *****

```

```

Columns . . . :   1  80                                Browse                                GOPZLIB/QDDSSRC
SEU==>
FMT PF .....A.....T.Name+++++RLen++TDpB.....Functions+++++
***** Beginning of data *****
0002.00          R EMPREC                                241223
0003.00          EMPNO              10  0                    241223
0004.00          EMPNAME              20                      241223
0005.00          EMPCITY              20                      241223
0006.00          DEPTNO              10  0                    241225
***** End of data *****

```

```

Columns . . . :   1  80                                Browse                                GOPZLIB/QDDSSRC
SEU==>
FMT LF .....A.....T.Name+++++.Len++TDpB.....Functions+++++
***** Beginning of data *****
0001.00          R EMPREC                                PFILE(EMPPF)          250102
0003.00          K EMPNO                                250113
***** End of data *****

```

```

0001.00 FEXTEMPPF  UF A E          K DISK
0002.00 FSALPF     IF  E          K DISK
0003.00 FEMPLFENO IF  E          K DISK
0004.00 DENO              S          10  0
0005.00 *
0006.00 C          READ      EEMPLFENO
0007.00 C          EVAL      ENO = EMPNO
0008.00 C          DOW      NOT %EOF(EMPLFENO)
0009.00 C      empno      setll  SALPF
0010.00 C      empno      reade  SALPF
0011.00 C          DOW      NOT %EOF(SALPF)
0012.00 C          IF      EMPNO = ENO
0013.00 C          EVAL      EMPID = EMPNO
0013.01 C          EVAL      ENAME = EMPNAME
0014.00 C          EVAL      MONTH = SMONTH
0015.00 C          EVAL      SALARY = SALAMT
0016.00 C          WRITE     EXTEMPREC
0017.00 C          ENDIF
0018.00 C      empno      READE  SALPF

```

```

0019.00 C          ENDDO
0020.00 C          READ      EMPLFENO
0021.00 C          EVAL      ENO = EMPNO
0022.00 C          ENDDO
0023.00 C          EVAL      *INLR = '1'
          ***** End of data *****

```

8. Create a new rpgle program to get employee no and input and show details of salary taken by employee on subfile screen it should have position to ,

```

0001.00 HOPTION(*NODEBUGIO:*SRCSTMT)
0001.01 *
0002.00 FEXTEMPPF IF E          K DISK
0003.00 FEMPSALSUB CF E          WORKSTN SFILE(EMPSFL:RRN) INFDS(DS1)
0004.00 *
0004.01 DRRN          S          4 0
0005.00 DDS1          DS
0006.00 DSUB1          378      379I 0
0007.00 *
0008.00 *****
0009.00 C          EVAL      RCD_NBR = 1
0010.00 C          DOW      *IN03 = '0'
0010.01 C          EXSR      CLRSFL
0010.02 C          EXSR      LOADSFL
0010.03 C          EXSR      DSPSFL
0010.04 C          EXSR      PROCSFL
0011.00 C          ENDDO
0012.00 C          EVAL      *INLR = '1'
0013.00 *****

```

```

0014.00 * CLEAR SUBFILE
0015.00 *****
0016.00 C      CLRSFL      BEGSR
0016.01 C          EVAL      *IN93 = '1'
0016.02 C          WRITE      EMPCTL
0016.03 C          EVAL      *IN93 = '0'
0017.00 C          ENDSR
0019.00 *****
0020.00 * LOAD SUBFILE
0021.00 *****
0022.00 C      LOADSFL      BEGSR
0023.00 C          EVAL      RRN = 0
0024.00 C      SEMPNO      SETLL      EXTEMPPF
0024.01 C      SEMPNO      READE      EXTEMPPF
0024.02 C          DOW      NOT %EOF()
0024.03 C          EVAL      RRN = RRN + 1
0024.04 C          WRITE      EMPSFL
0024.05 *
0024.06 C          IF      SPOS = MONTH
0024.07 C          EVAL      RCD_NBR = RRN

```

```

0024.08 C          ENDIF
0024.09 *
0024.10 C      SEMPNO      READE      EXTEMPPPF
0024.11 C          ENDDO
0026.00 C          ENDSR
0027.00 *****
0028.00 * DISPLAY SUBFILE
0029.00 *****
0030.00 C      DSPSFL      BEGSR
0030.01 C          WRITE      FOOTER
0031.00 C          EVAL      *IN92 = '1'
0032.00 C          IF      RRN > 0
0033.00 C          EVAL      *IN91 = '1'
0033.01 C          EVAL      *IN94 = '1'
0033.02 C          ENDIF
0033.03 C          EXFMT      EMPCTL
0033.04 C          IF      SUB1> 0
0033.05 C          EVAL      RCD_NBR = SUB1
0033.06 C          ELSE
0033.07 C          EVAL      RCD_NBR = 1

```

```

0033.08 C          ENDIF
0033.09 C          EVAL      *IN92 = '0'
0033.10 C          EVAL      *IN91 = '0'
0033.11 C          EVAL      *IN94 = '0'
0034.00 C          ENDSR
0034.01 *****
0034.02 * PROCESS SUBFILE
0034.03 *****
0034.04 C      PROCSFL      BEGSR
0034.05 C          IF      *IN03 = '1'
0034.06 C          LEAVESR
0034.07 C          ENDIF
0034.08 C          IF      RRN = 0
0034.09 C          LEAVESR
0034.10 C          ENDIF
0034.11 C          ENDSR
***** End of data *****

```

EMPLOYEE SALARY DETAILS

WING15
1/14/25

ENTER EMPLOYEE NO : POSITION TO :

Employee_Name	Month	Salary
KEERTHI	JANUARY	0002900000
KEERTHI	FEBRUARY	0002900000
KEERTHI	MARCH	0002900000
KEERTHI	APRIL	0002900000

Bottom

F3=EXIT

POSITION TO : MARCH

EMPLOYEE SALARY DETAILS

WING15
1/14/25

ENTER EMPLOYEE NO : 105 POSITION TO : MARCH

Employee_Name	Month	Salary
KEERTHI	JANUARY	0002900000
KEERTHI	FEBRUARY	0002900000
KEERTHI	MARCH	0002900000
KEERTHI	APRIL	0002900000

Bottom

F3=EXIT

EMPLOYEE SALARY DETAILS

WING15
1/14/25

ENTER EMPLOYEE NO : POSITION TO :

Employee_Name	Month	Salary
KEERTHI	MARCH	0002900000
KEERTHI	APRIL	0002900000

Bottom

F3=EXIT