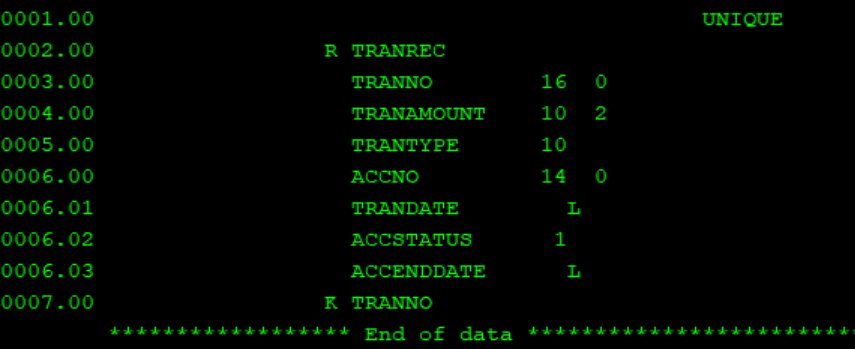
RPGLE ASSIGNMENT (11.1.2025)

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.





1. 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)

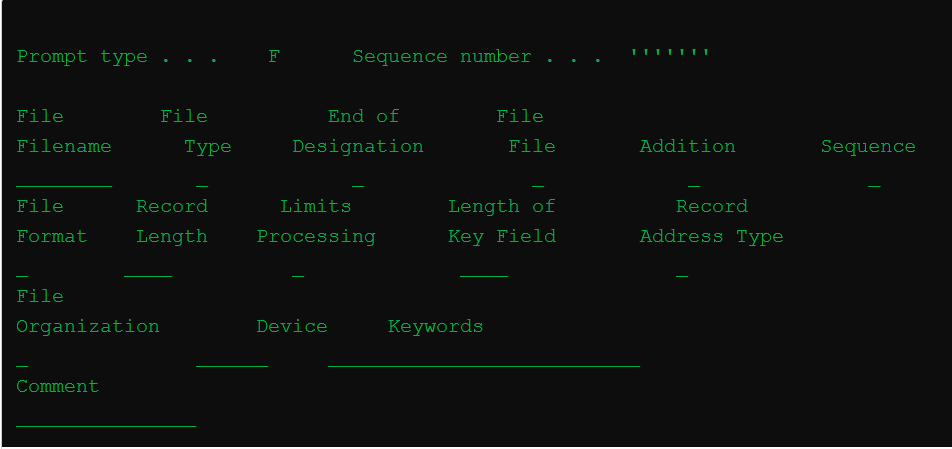
\*USA (mm/dd/yyyy)

\*EUR (dd.mm.yyyy)

e.g. H datfmt(\*MDY)

By default FMT will be \*ISO.

1. F SPEC (File description)



**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

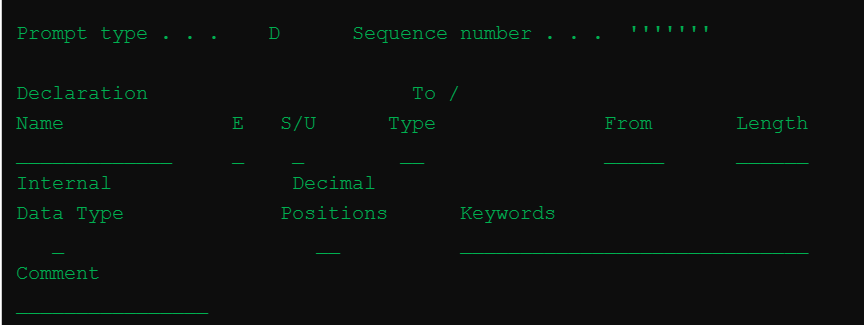
INFSR - FOR FILE LEVEL. INFSR IS A KEYWORD NOT A SUBROUTINE UNLIKE \*PSSR.

FFILENAME IF E A DISK INFSR(SR)

SR SHOULD GET EXECUTED WHEN THERE IS FILE ISSUE.

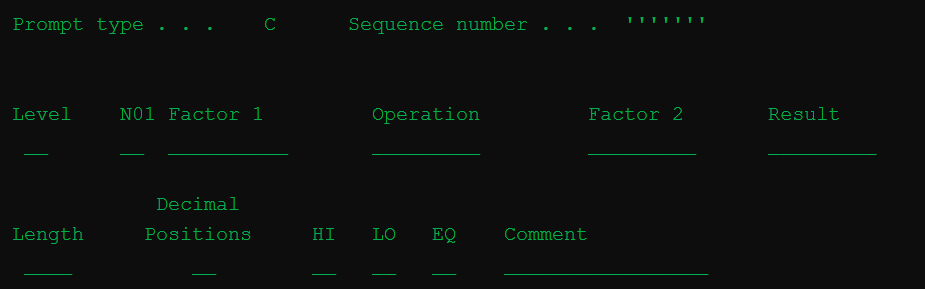
SR CAN BE REPLACED BY \*PSSR TOO.

1. D SPEC(Definition specifications)



**Keywords** : CONST(value), LIKE(RPG\_name), OCCURS, CTDATA, DIM, PERRCD(numeric\_constant), INZ

1. C SPEC (Calculation specifications)



**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 :

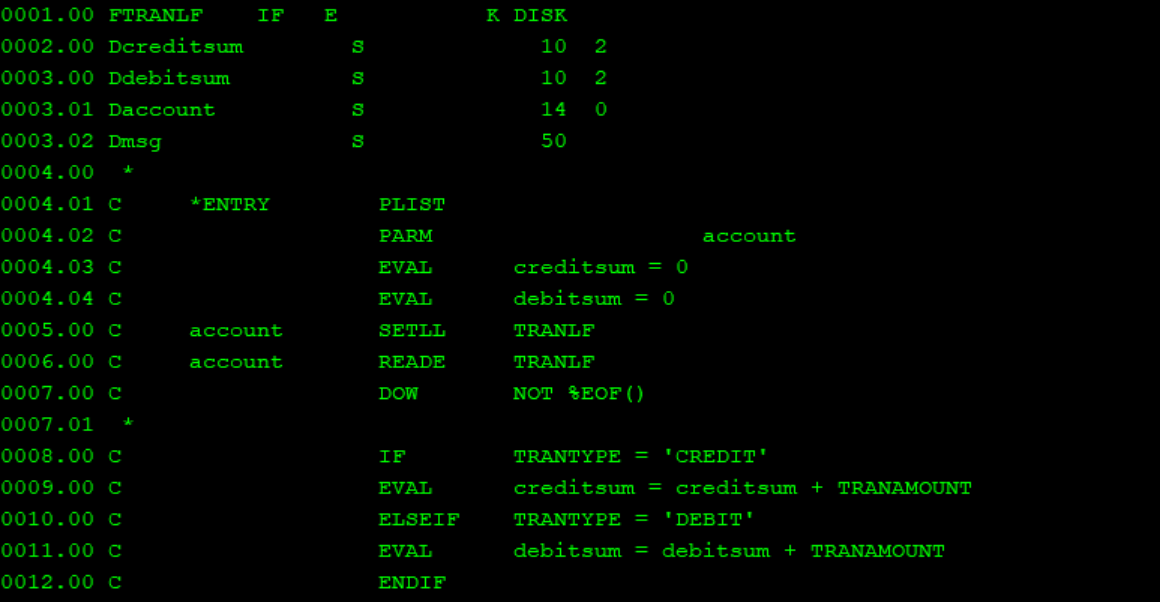
Decimal positions :

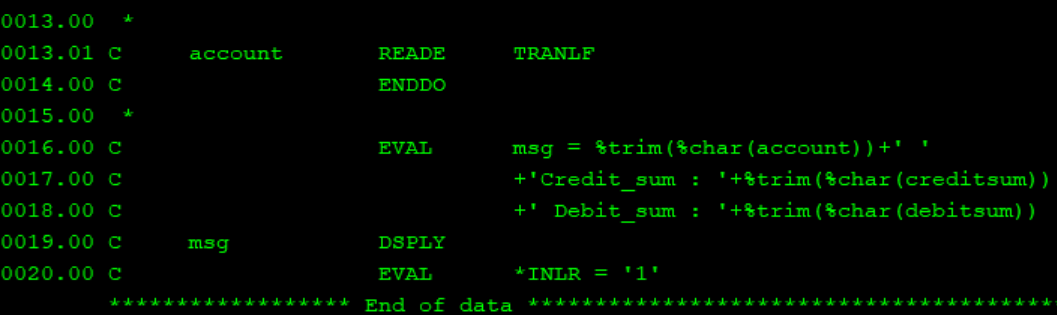
1. P SPEC(Procedure specifications) :

Deals with procedures of different languages (RPG ILE).

1. O SPEC (Output specifications) :

For output settings





RETURN VS INLR

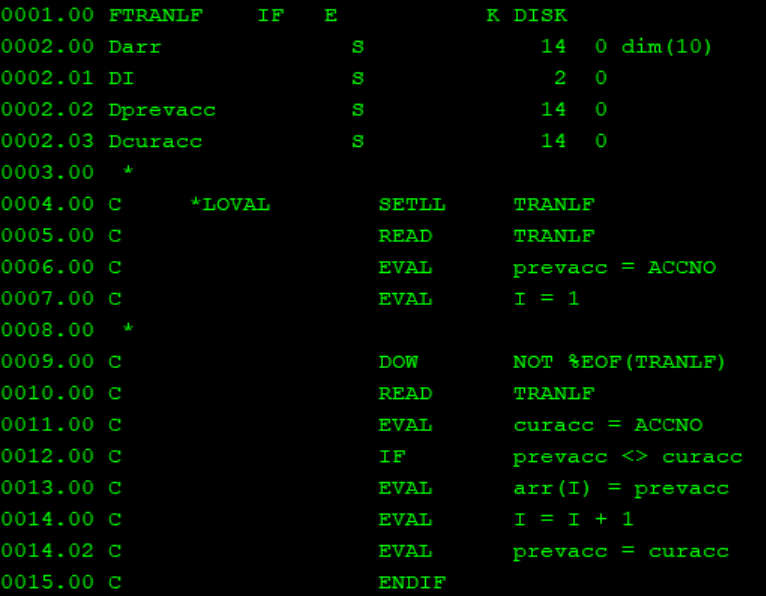
INLR => ALL THE FILES WILL BE CLOSED, CLEAR THE MEMORY.

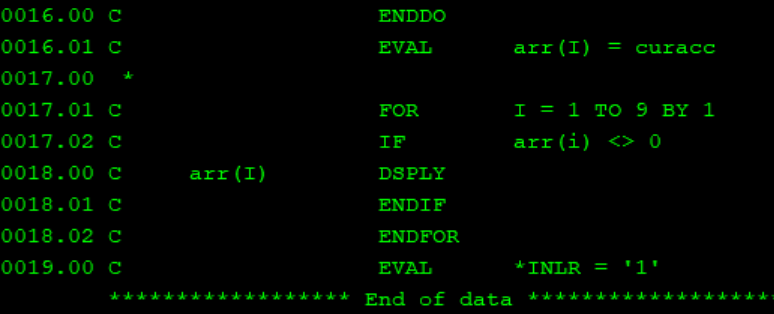
RETURN => CLOSING WILL NOT HAPPEN, VARIABLE VALUES WILL NOT BE CLEARED UNLIKE INLR. IT WILL CONFLICT WITH OTHER PROGRAM LIKE IF NEXT PROGRAM DONT HAVE INTIALIZATION FOR THOSE VARIABLES USING AGAIN, IT WILL RETAIN THE VALUE OF PREV PROGAM.

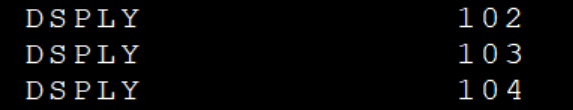
ADVANTAGE OF RETURN



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

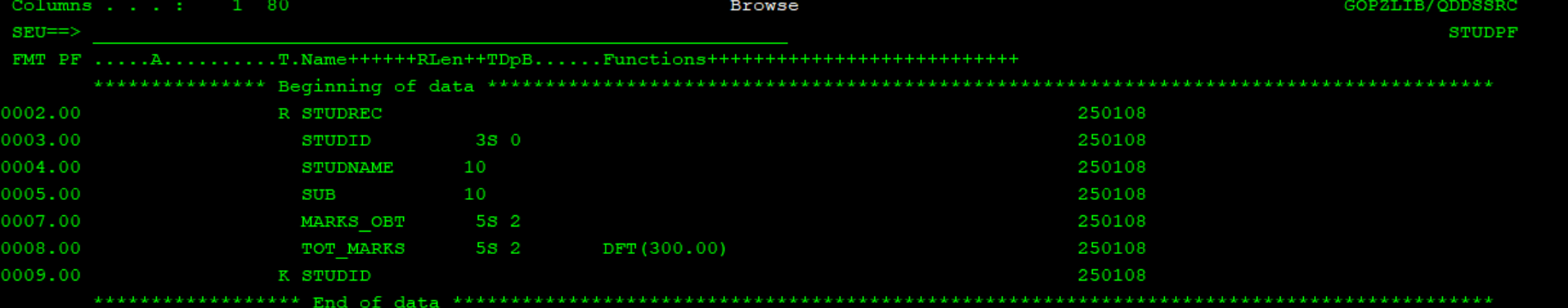


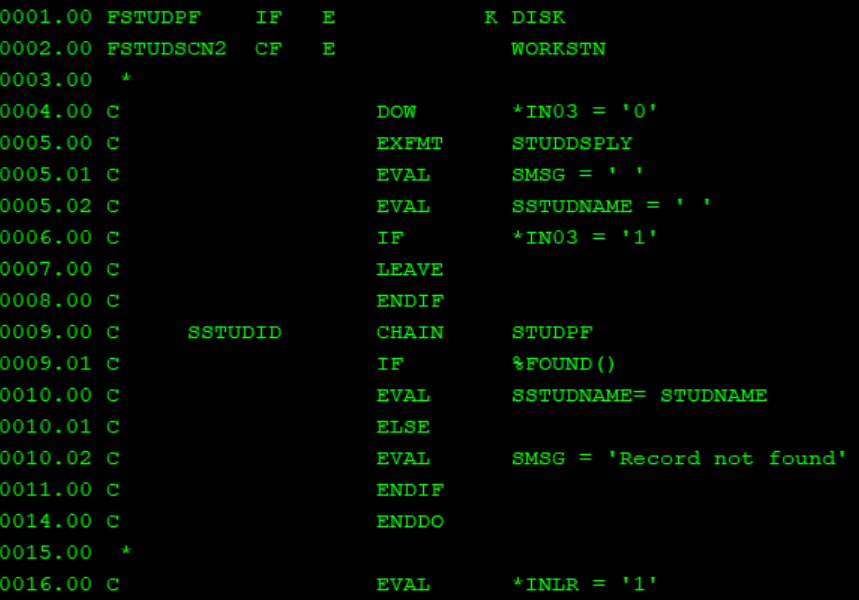


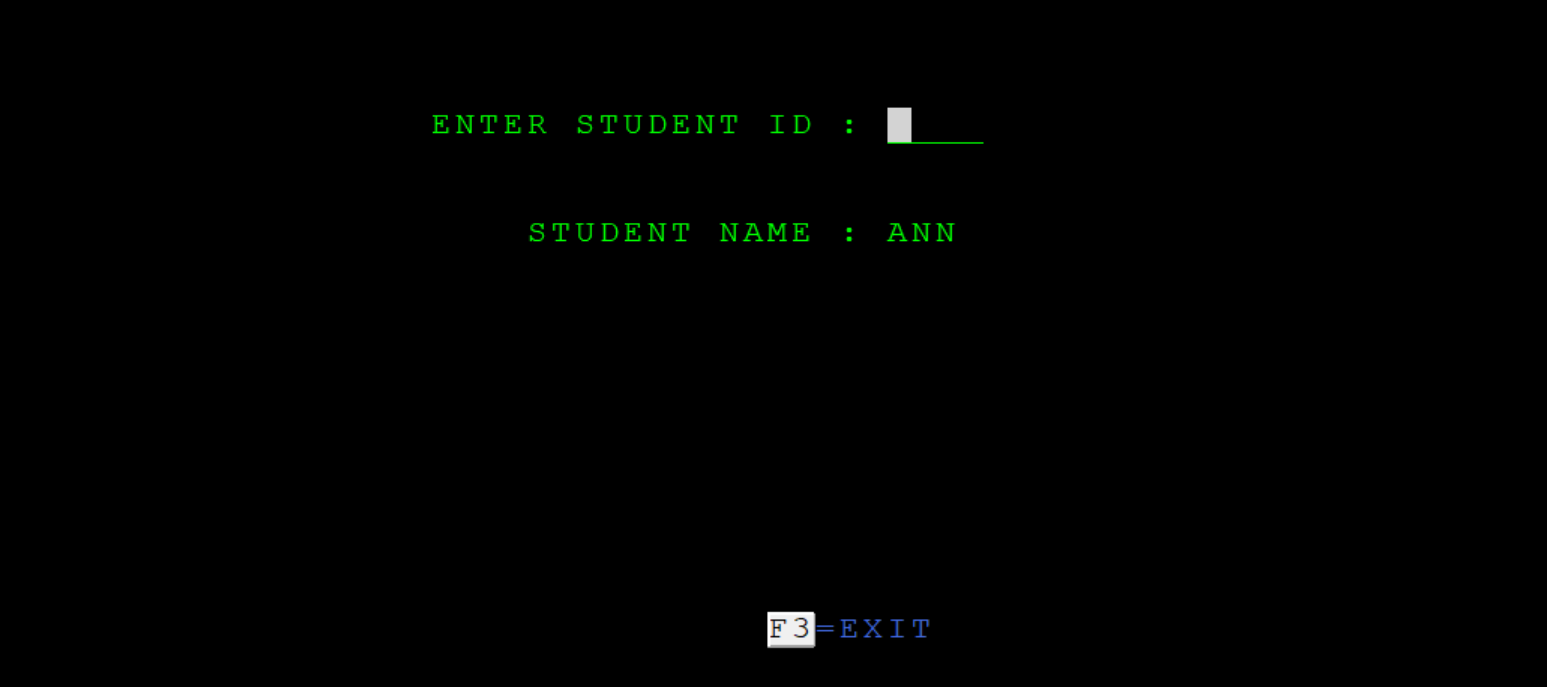


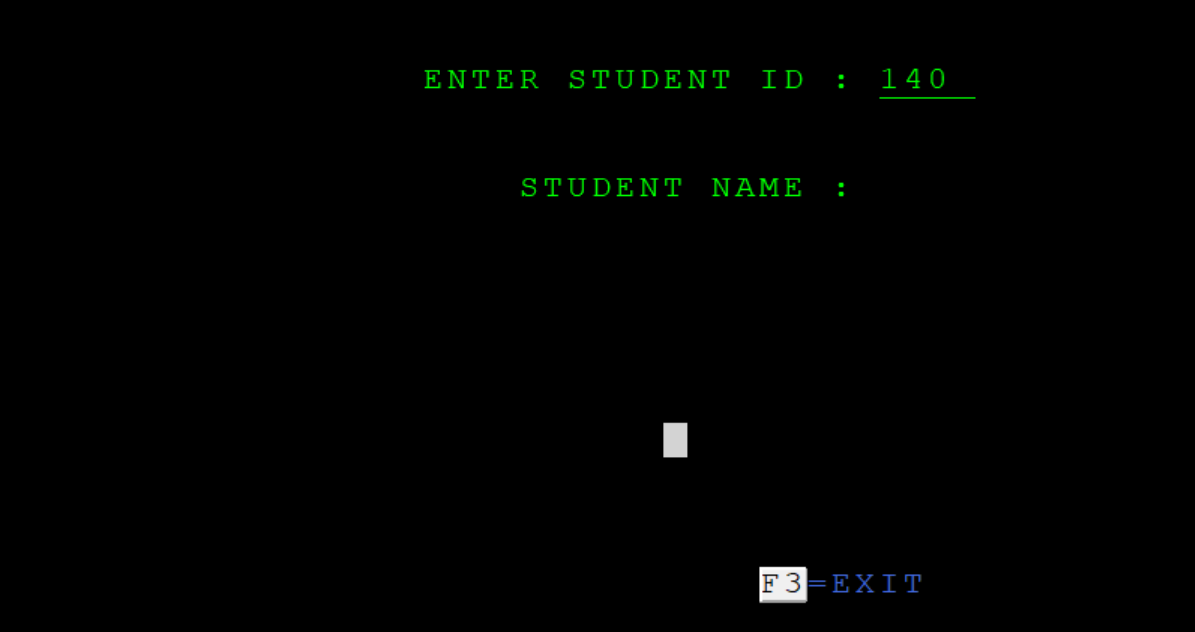
3. Create a rpgle program to take student id as input and display student name on screen

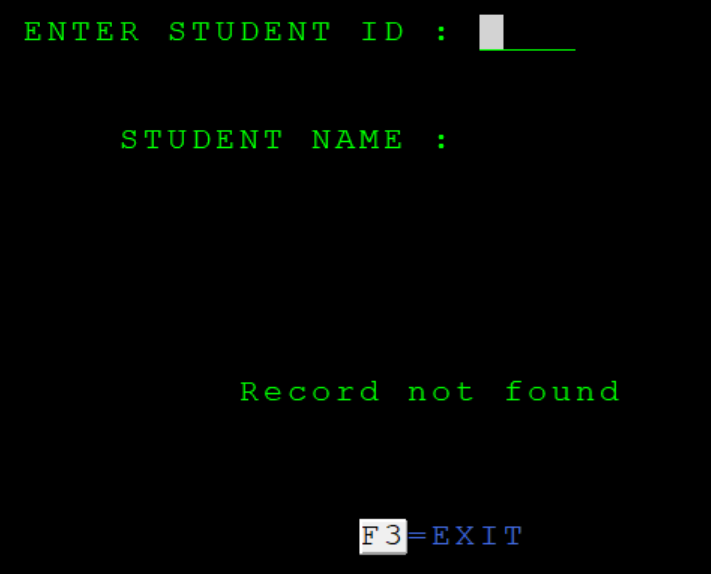
use studpf for this .







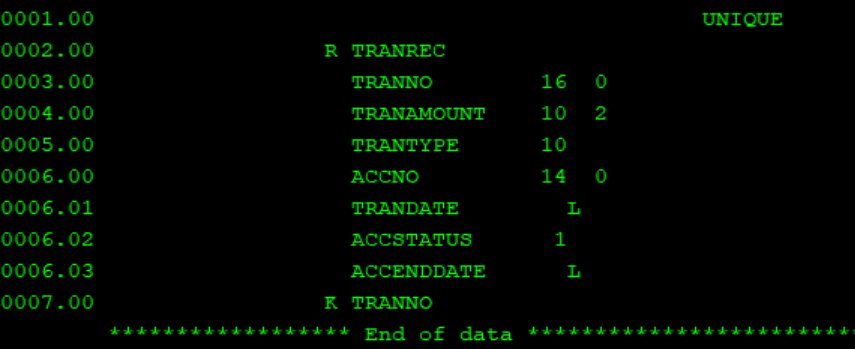




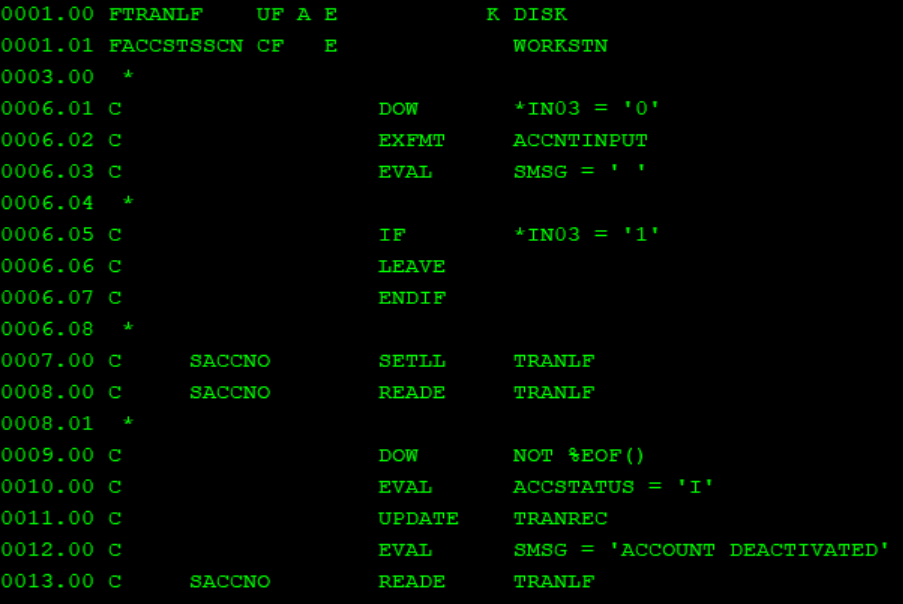
4. Add a new field as account status(char 1) in transaction pf.

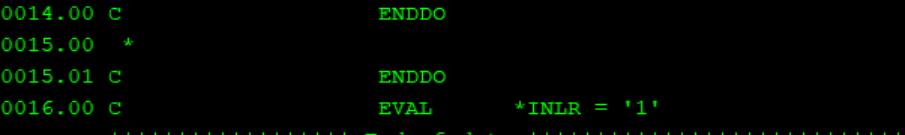
Create a screen to take account no as input and mark all records in transaction pf with satus 'I' for that account .

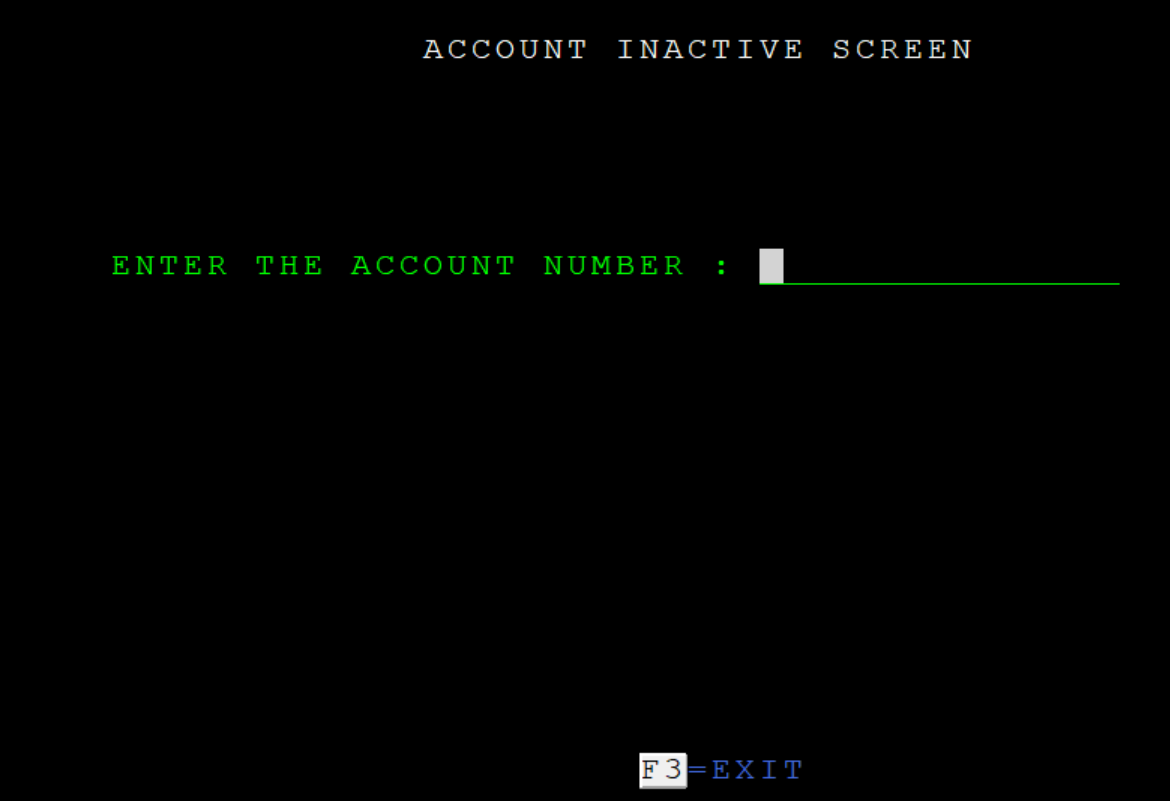
screen name can be : account inactivate screen.





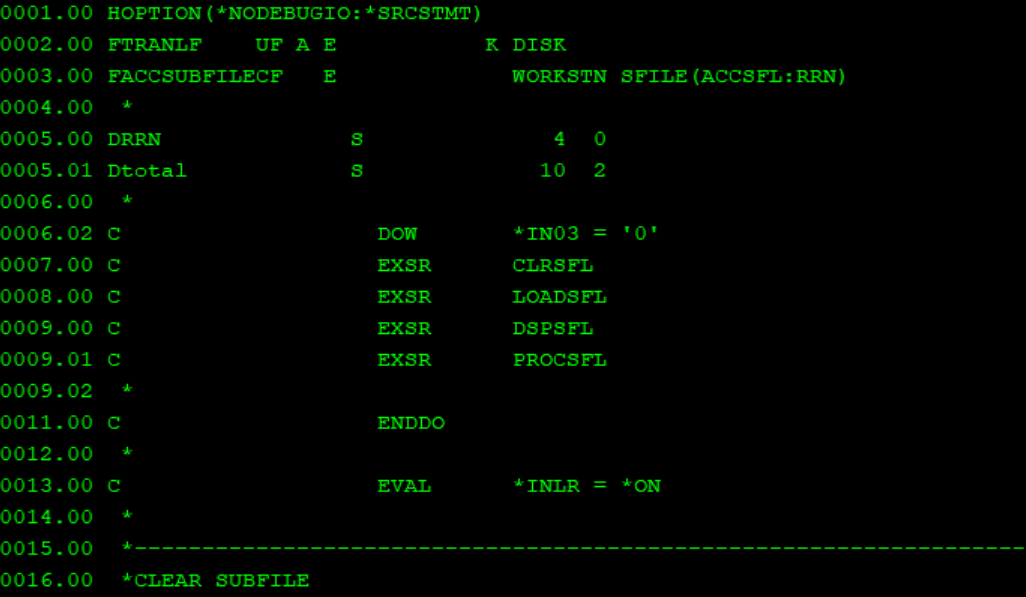




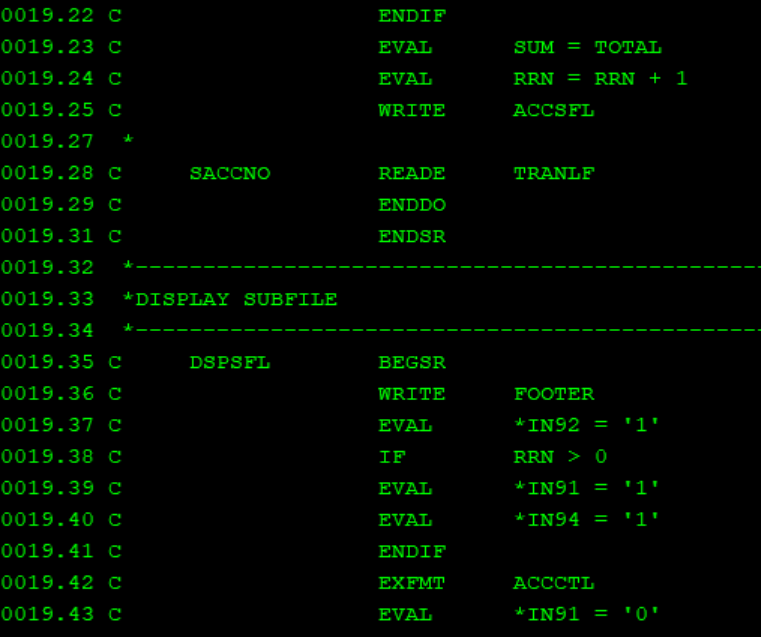


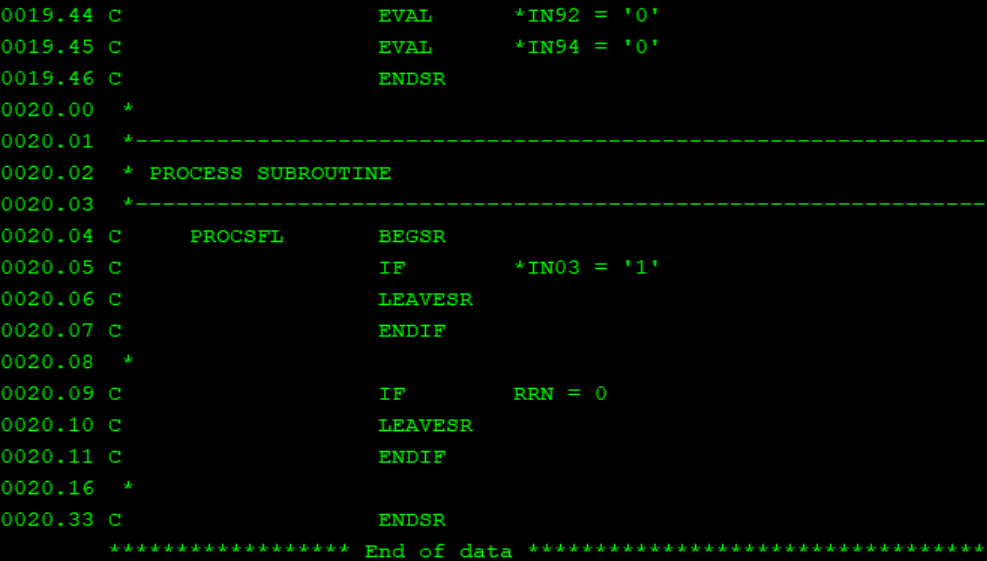
5. Create a rpgle program to take account as input and display all transaction of that account on screen.

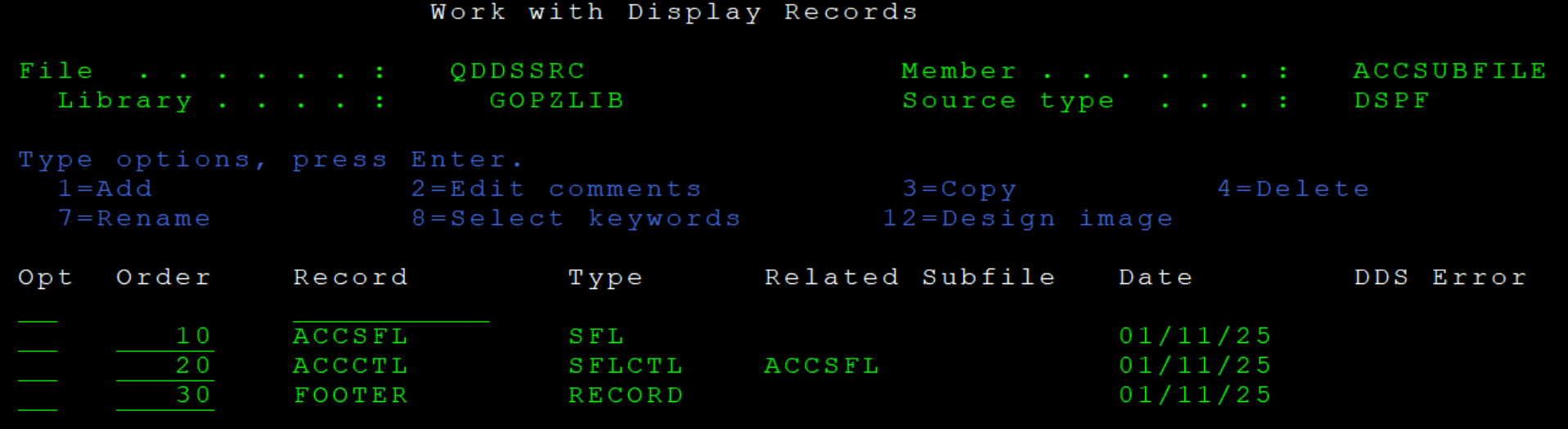
In footer it should show sum.

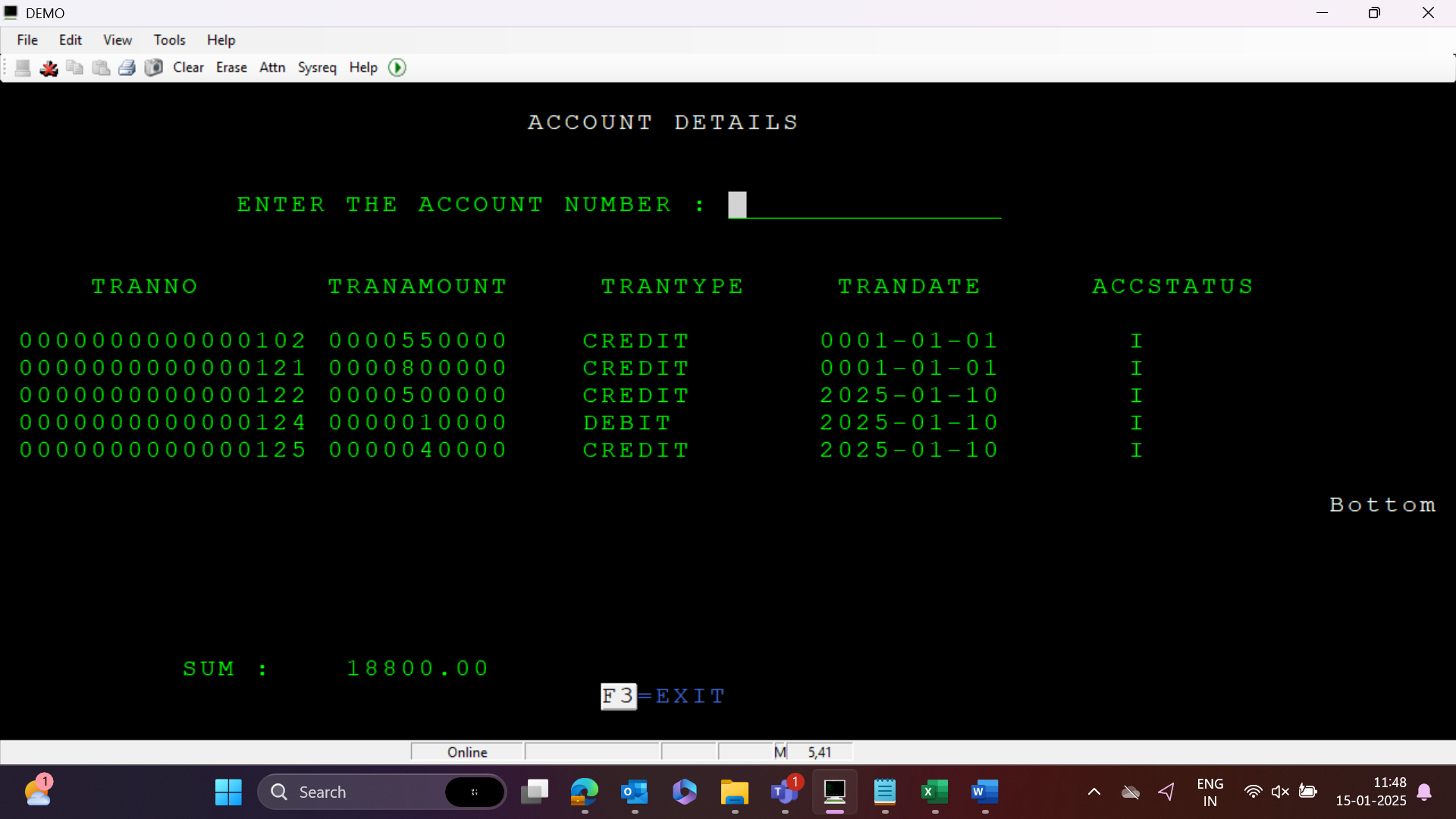




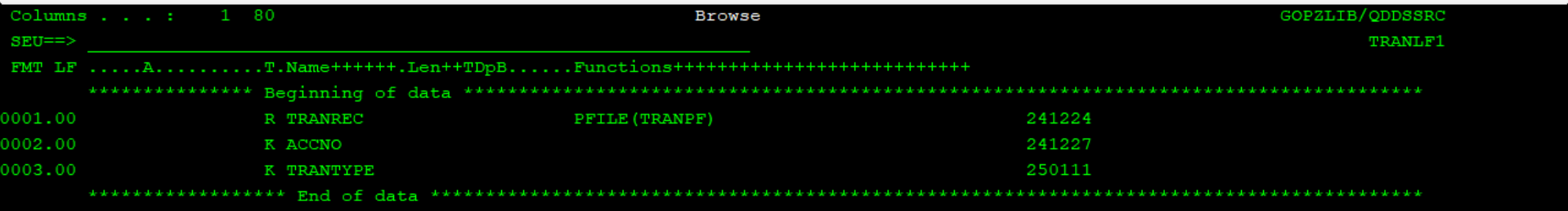




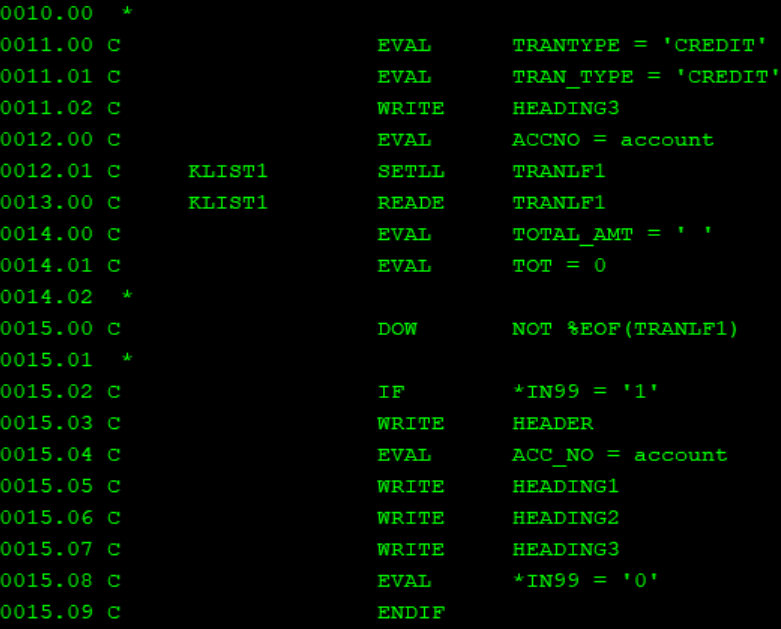


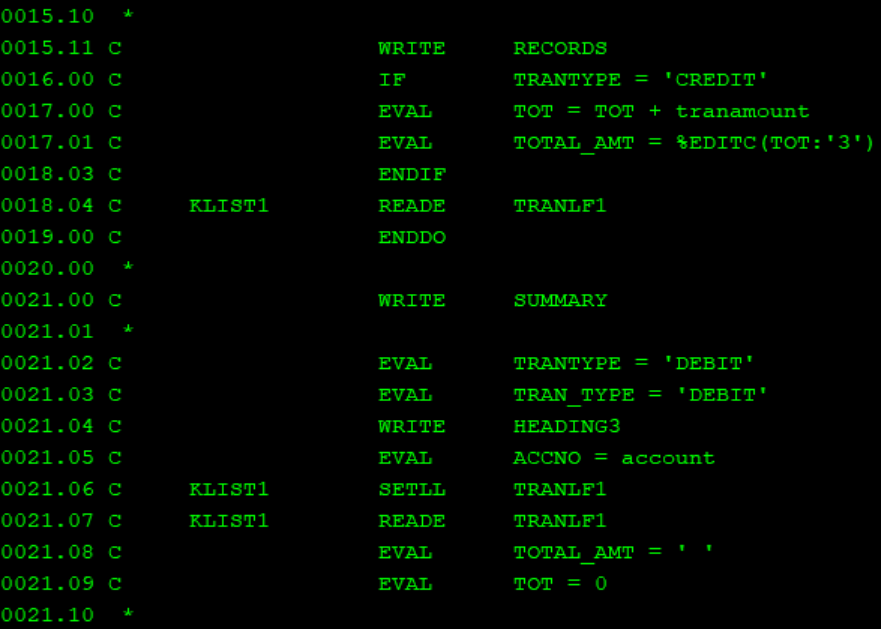


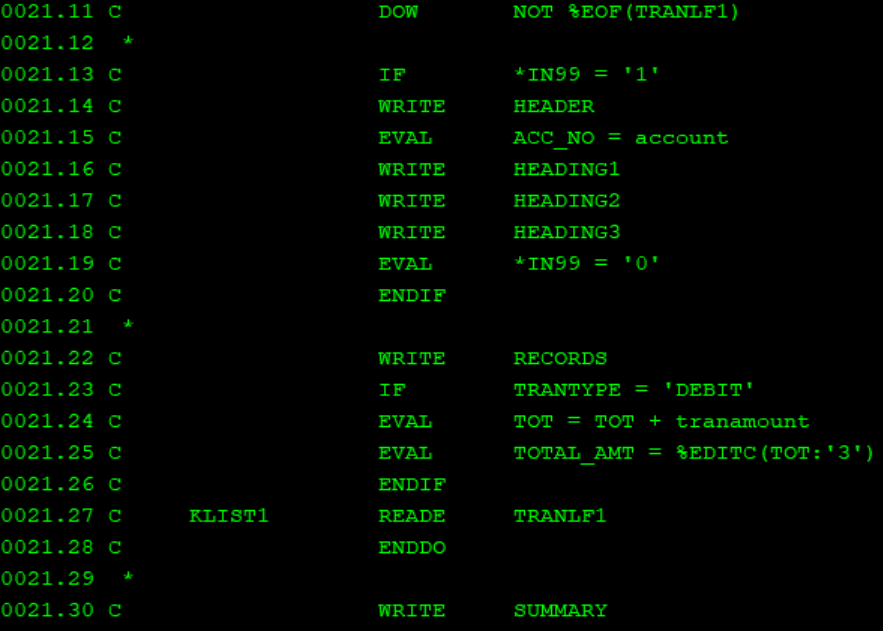
6. Create a rpgle program to take account no as input and create a report showing debit and credit transactions separately and sum of each type.

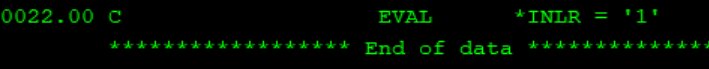


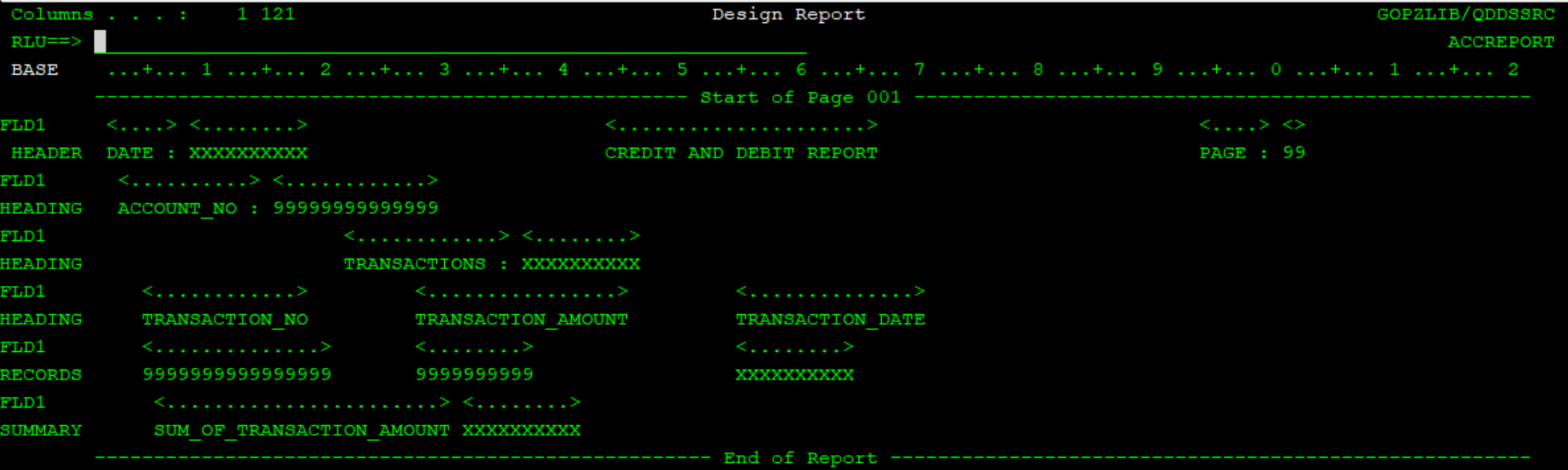


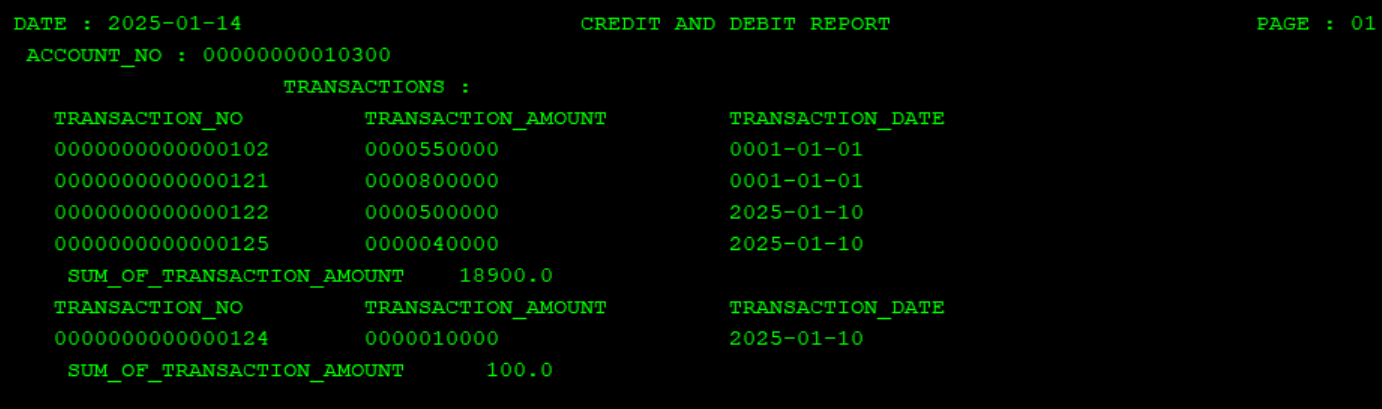








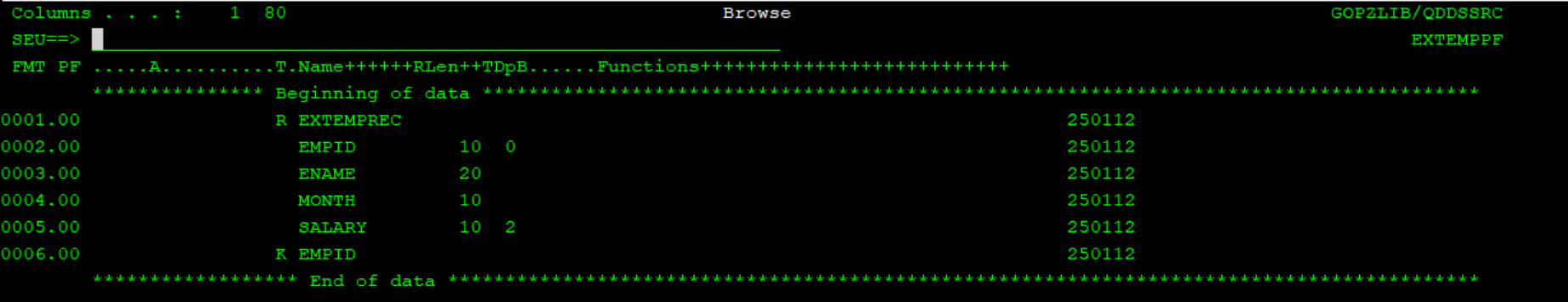


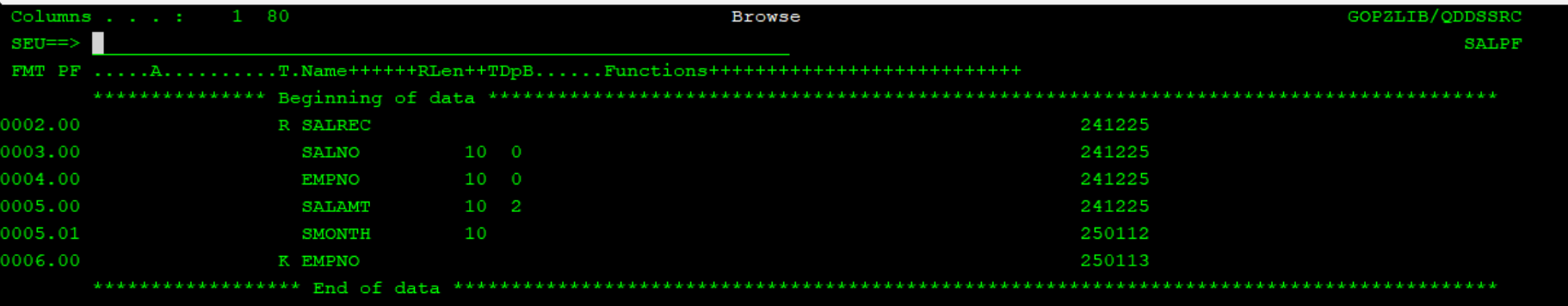


7. Create a new table employee salary extract.

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

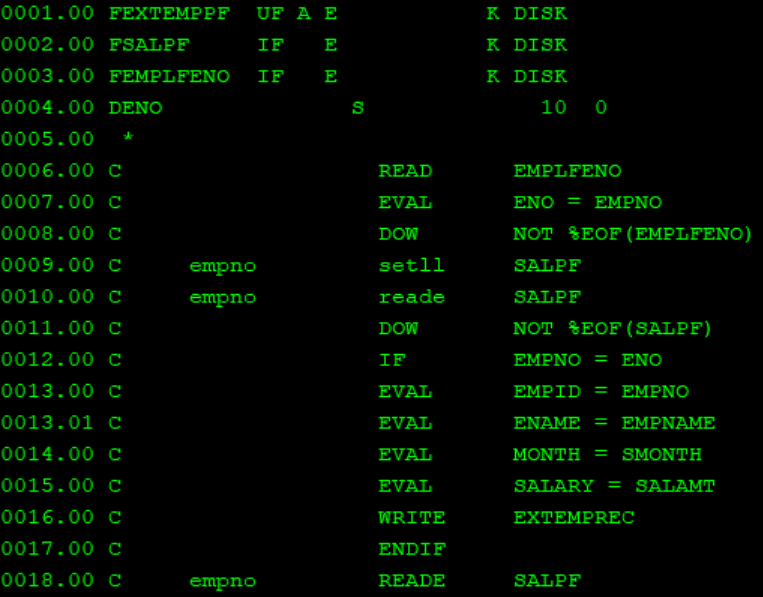
Create rpgle program to populate this file it should get details from employee pf and salary pf . It should populate details of all employees.

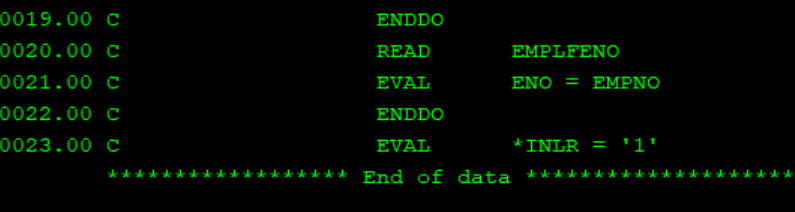












8. Create a new rpgle program to get employee no and input and show details of salary taken by employee on subfile screen it should have position to ,

