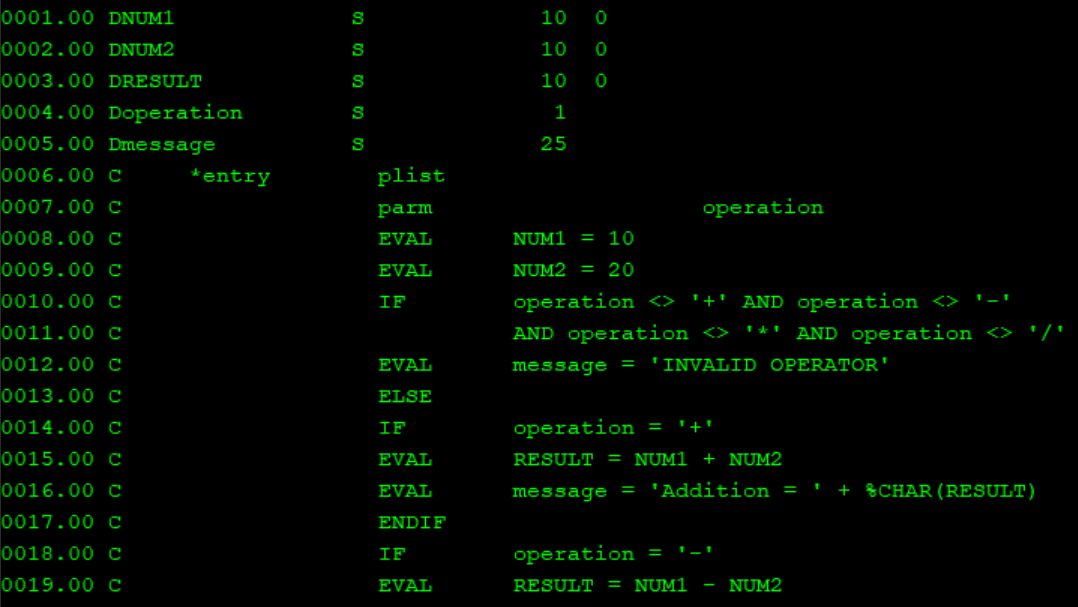
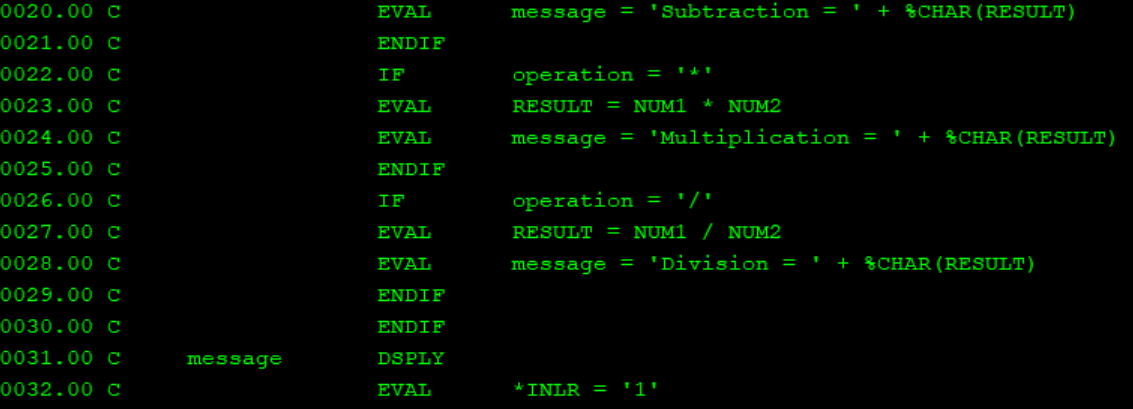
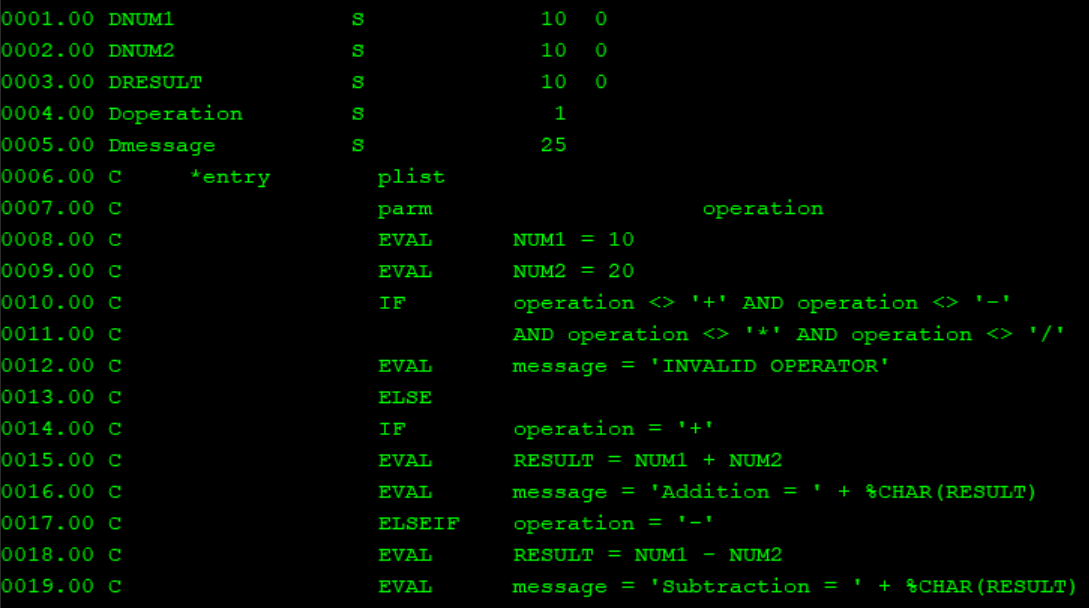
**RPGLE (Report Program Generator with Integrated Language Environment)**

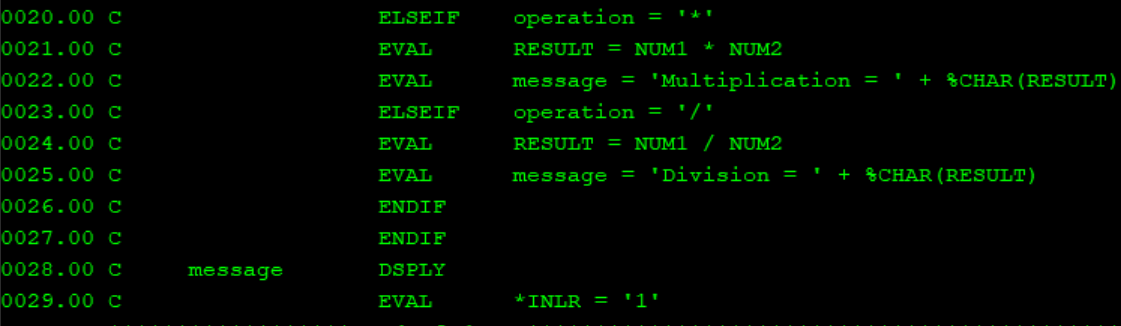
1. Arithmetic operations program using IF



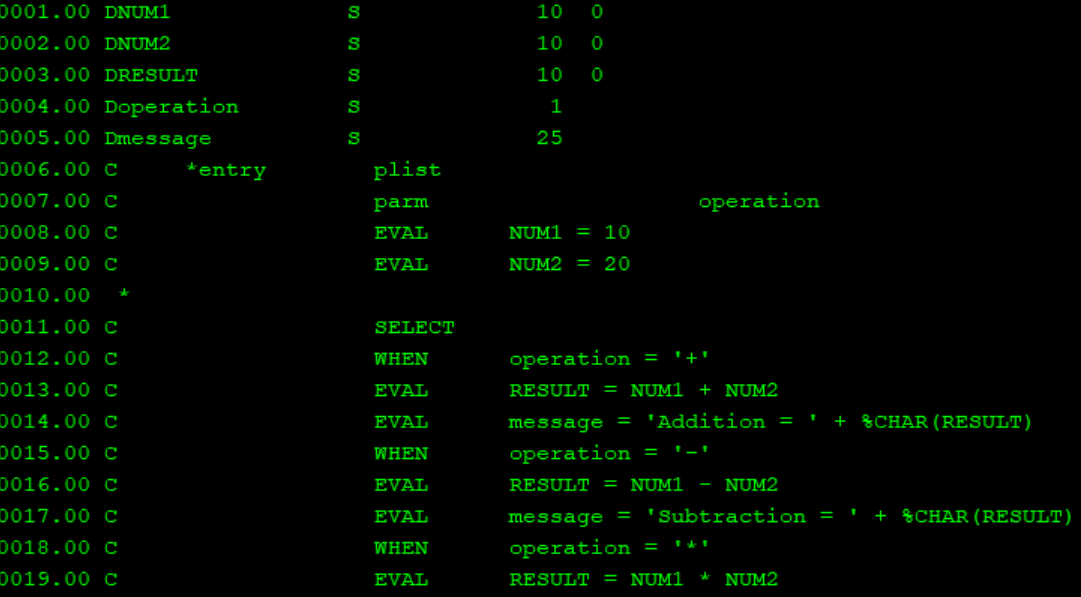


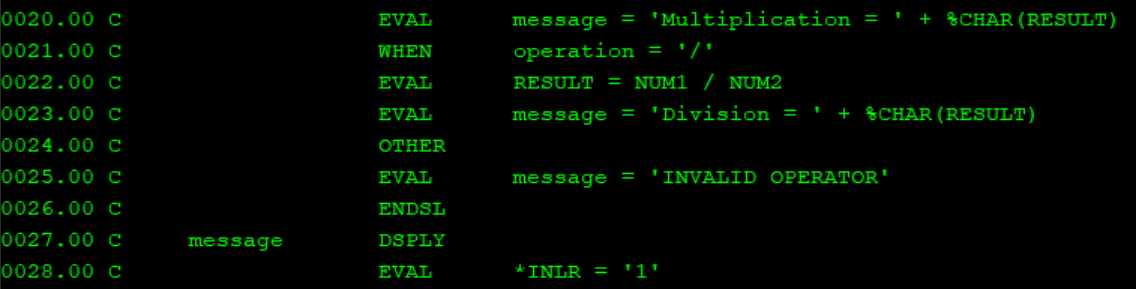
1. Arithmetic operations using IF-ELSE



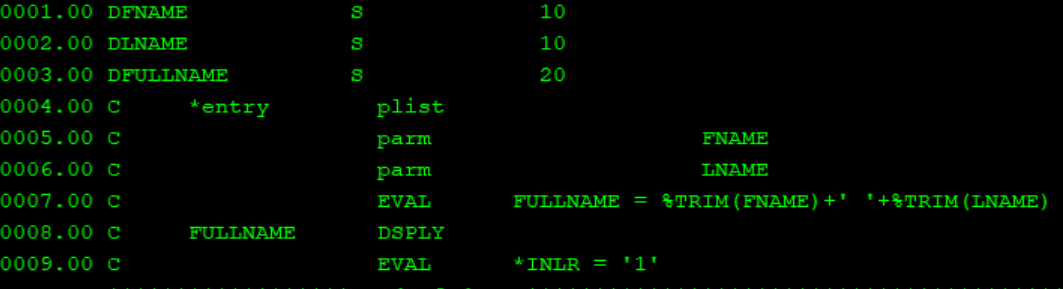


1. Arithmetic operations using SELECT WHEN

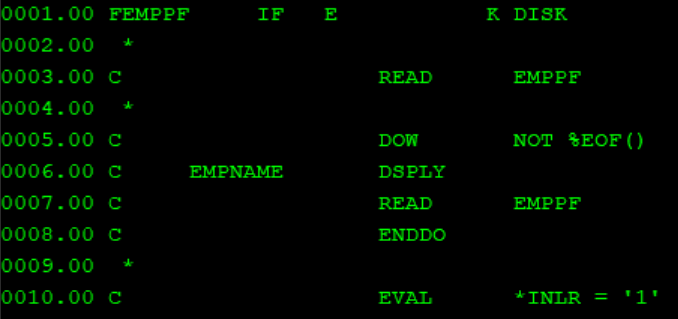




1. String concatenation program



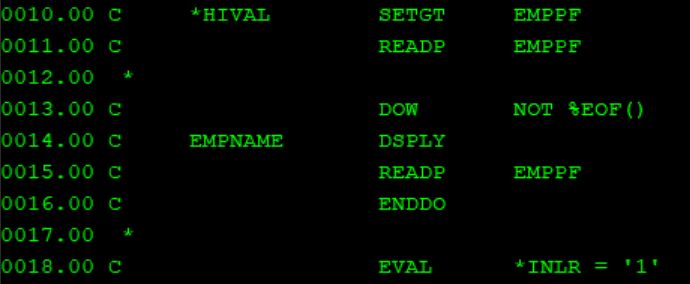
1. Reading file



1. Reading file using SETLL

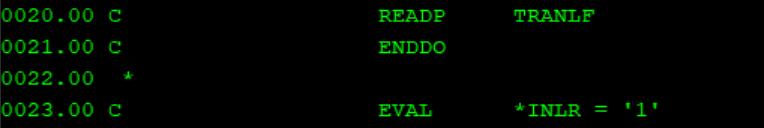


1. Reading file using SETGT



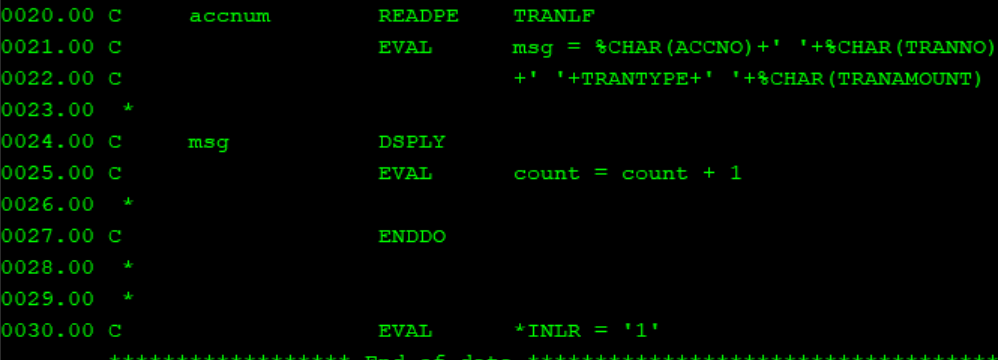
1. SETGT and READP



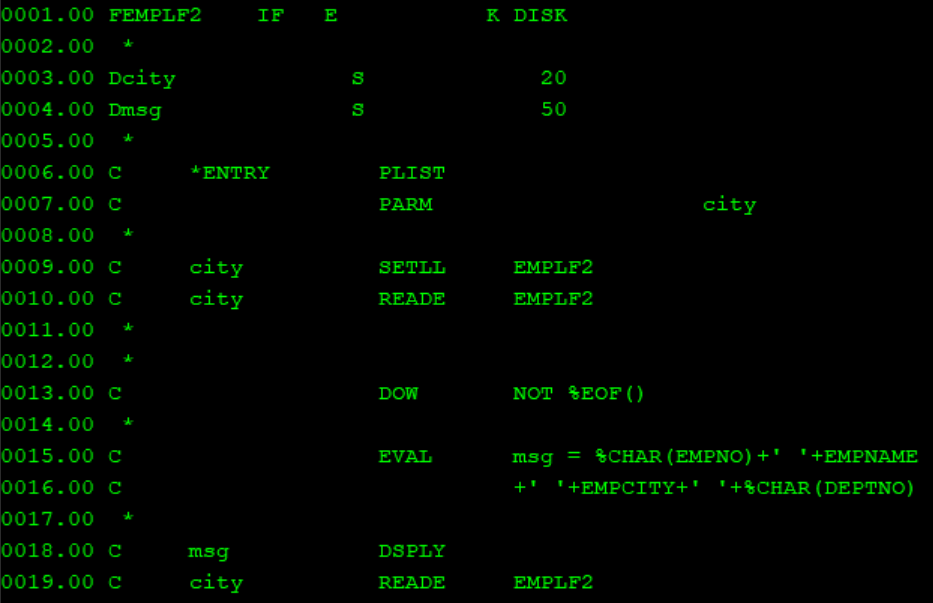


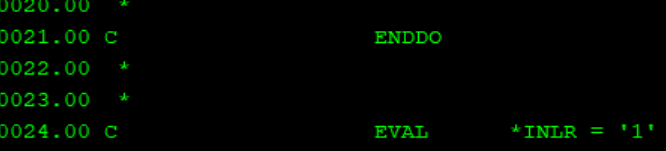
1. LEAVE and READPE



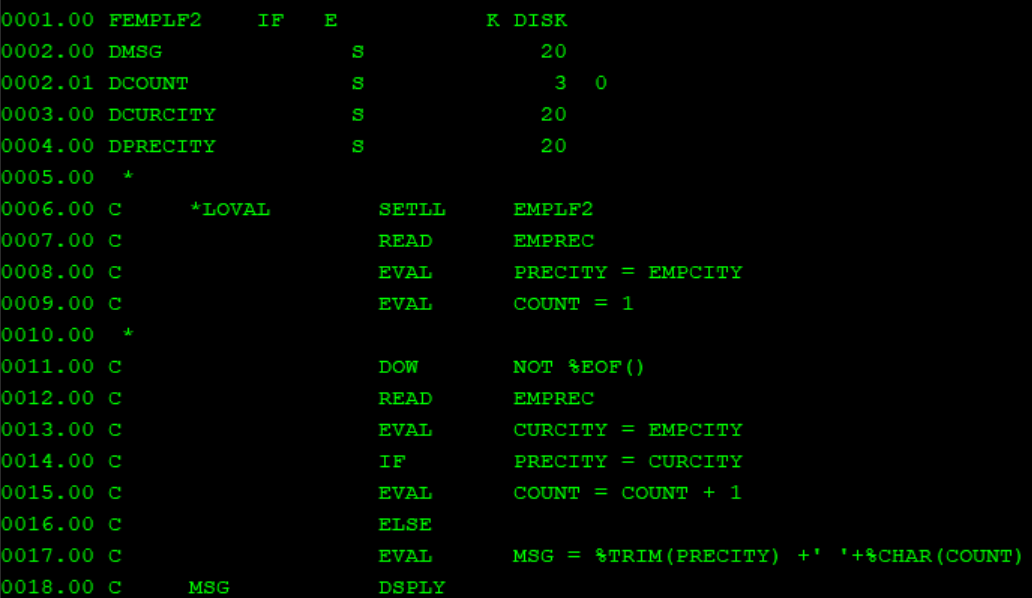


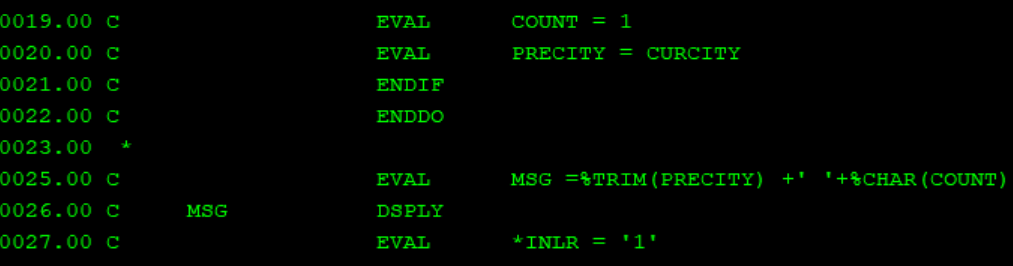
1. READE



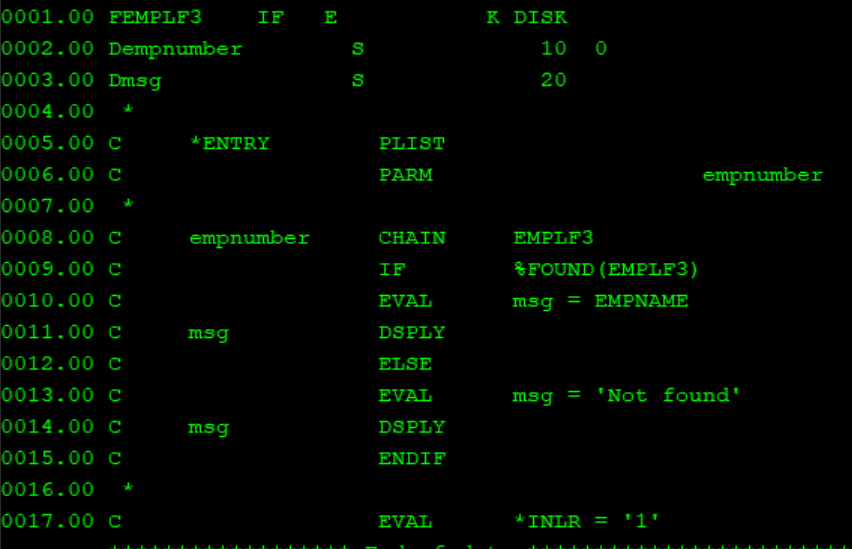


1. Employee city count

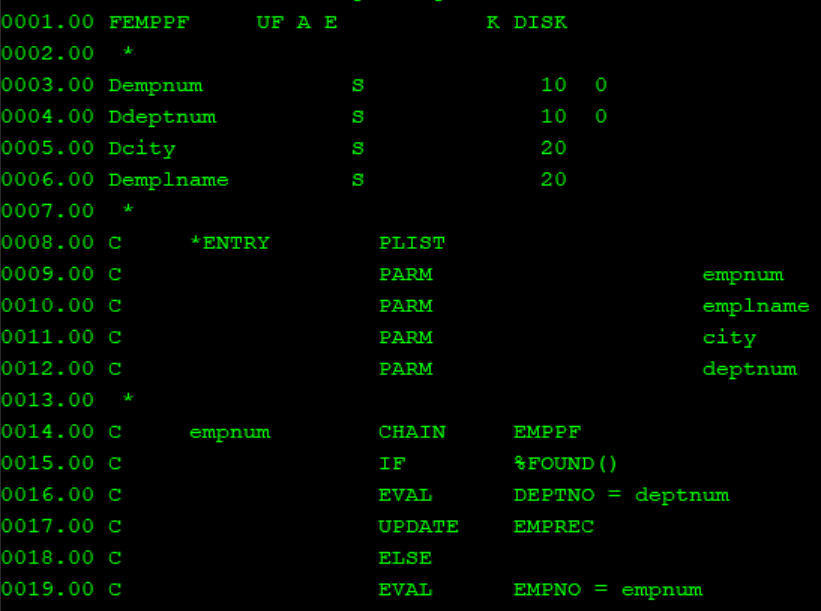


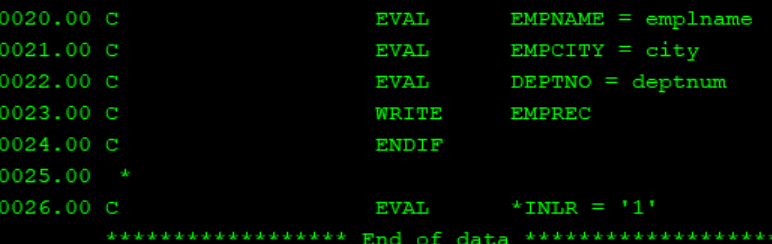


1. Program using chain

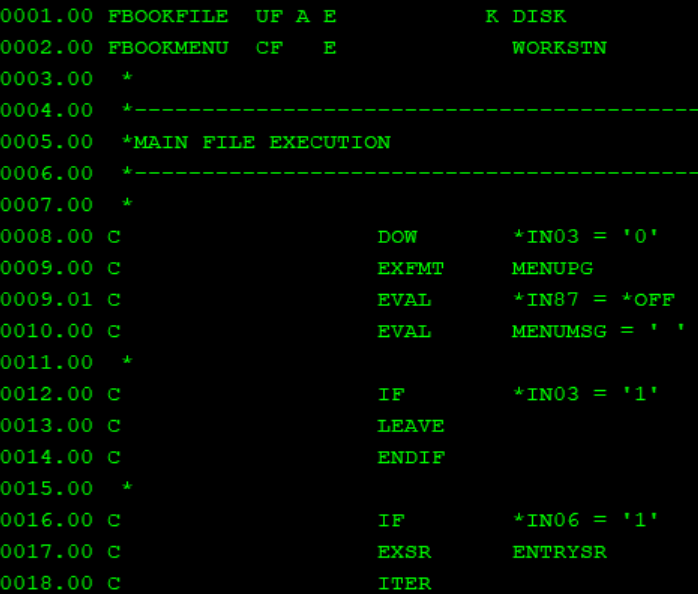


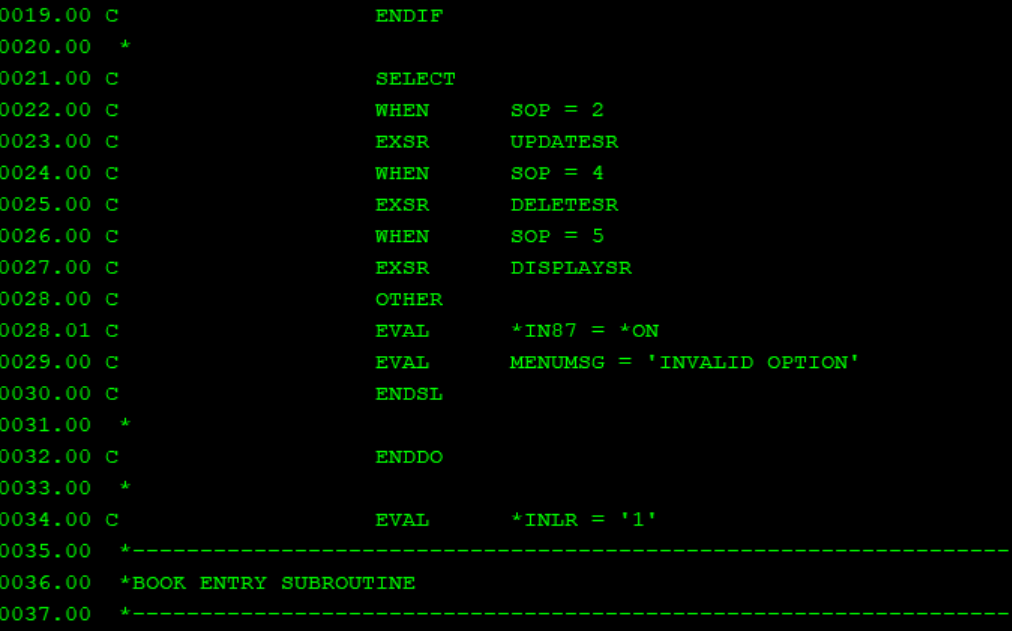
1. Write and update file

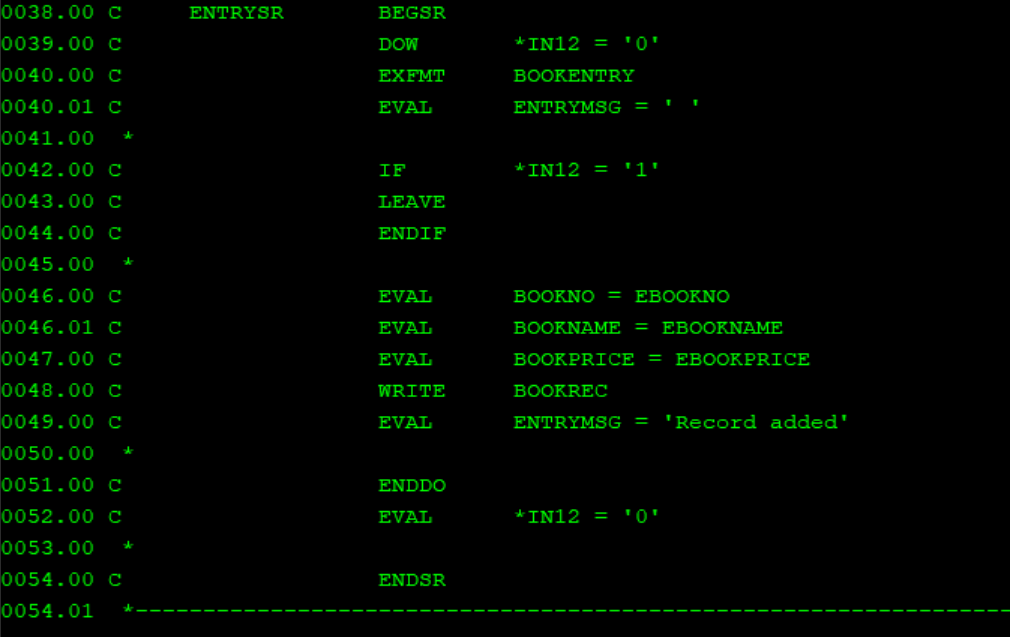




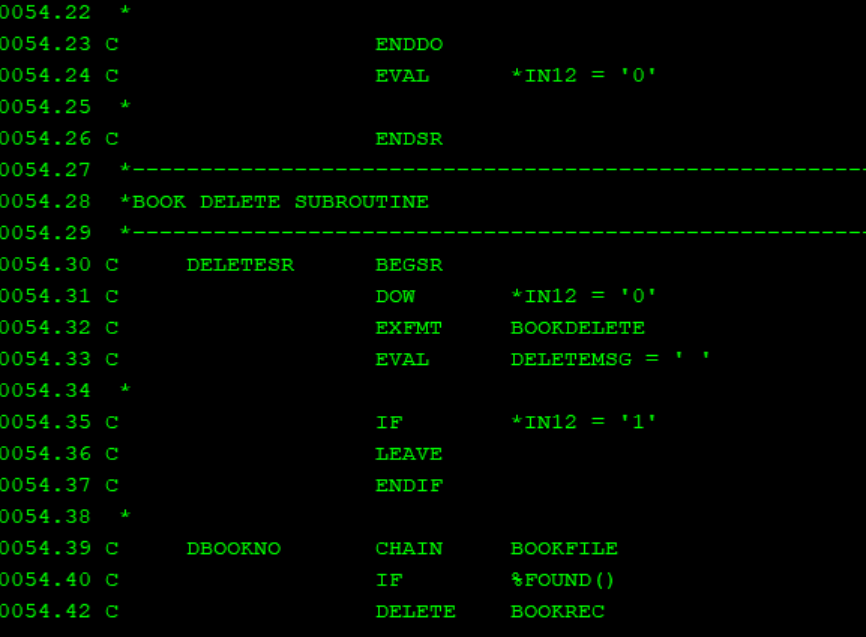
1. Menu program

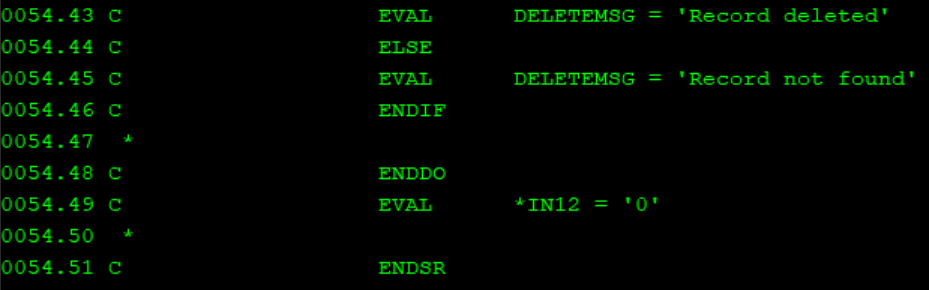


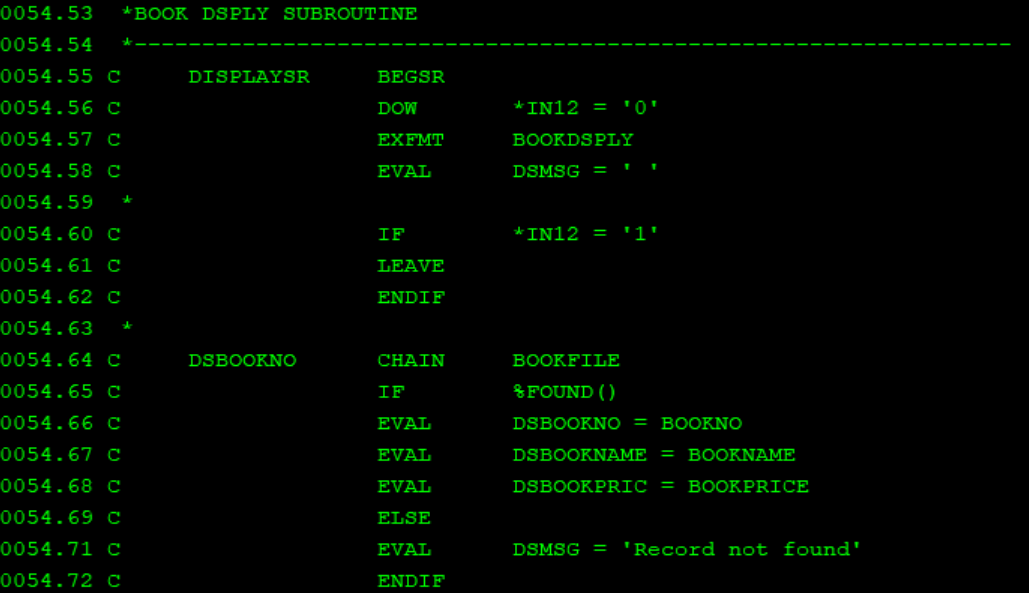


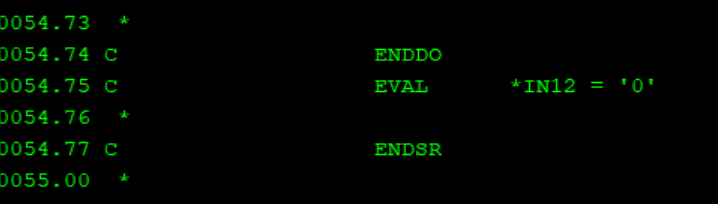












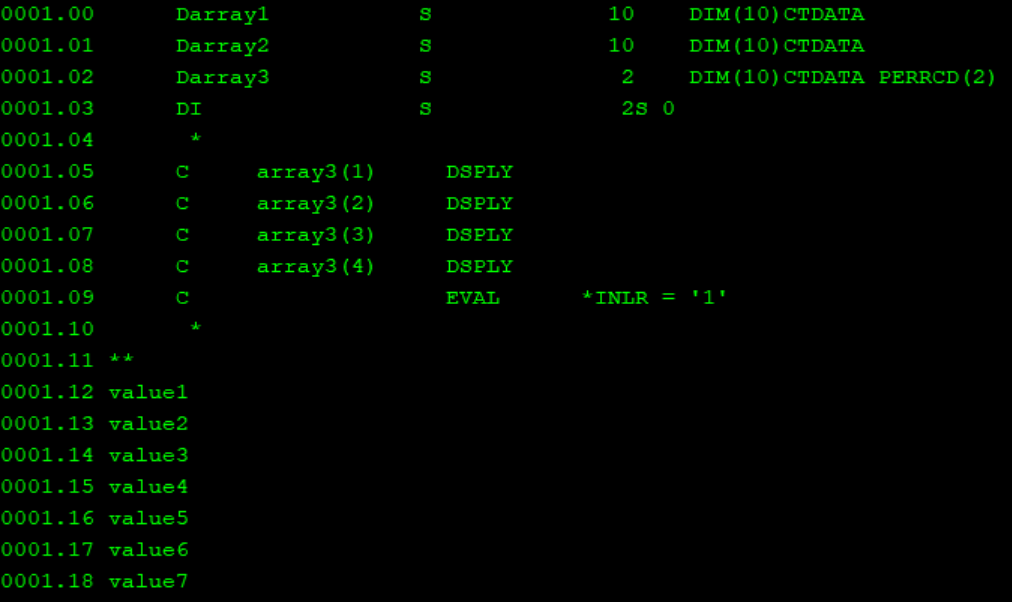
**Array programs**

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 ()

1. Compile time array example



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.

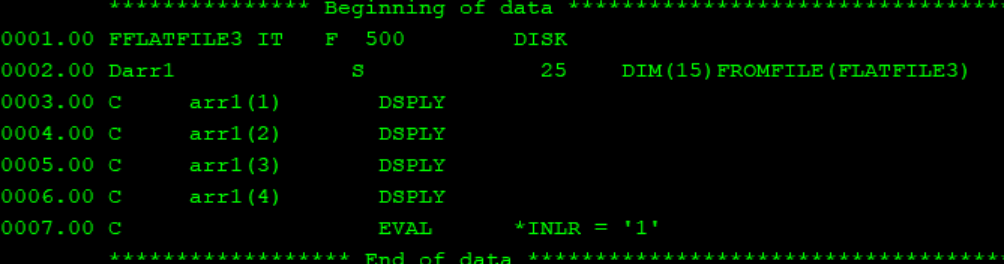


Flat files : Files without any structure

CRTPF COMMAND WITH RECORD length will create a flat file

CRTPF FILE(FLATFILE) RCDLEN(500)

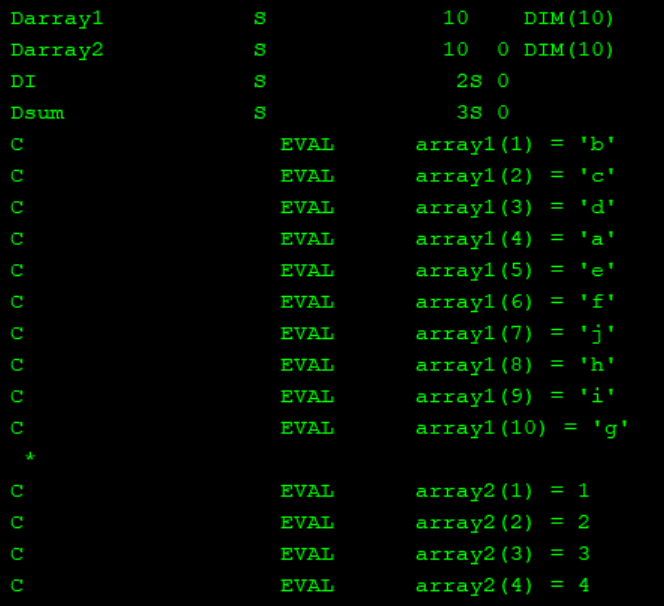
To view flatfiles : DSPPFM FILE(EASYCLASS1/FLATFILE1)

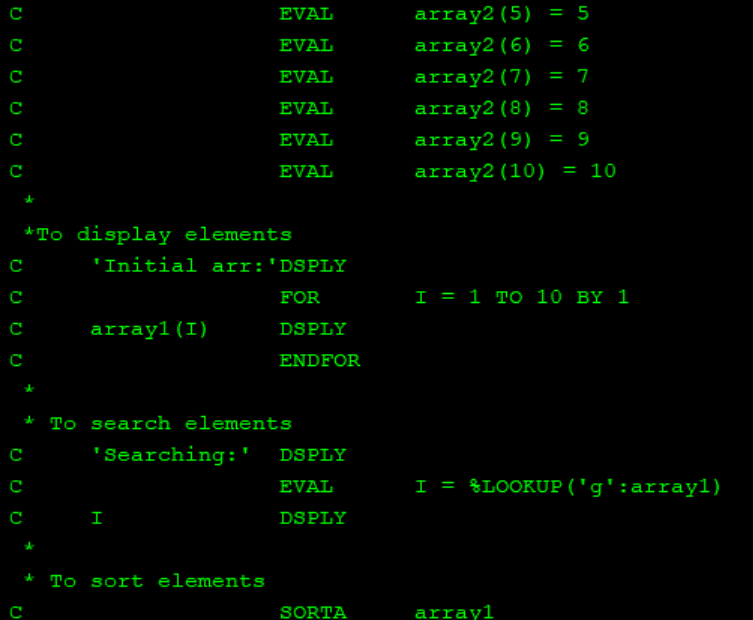


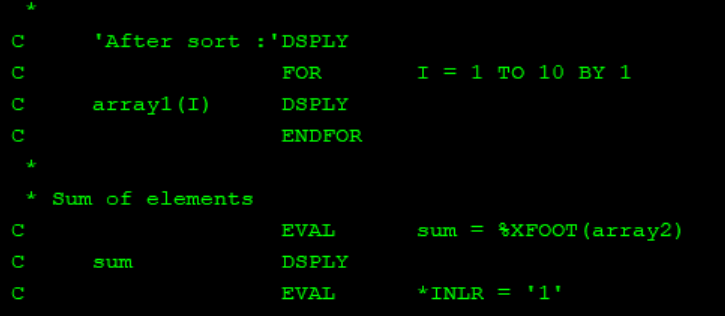
3) Run time array

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

1. Runtime array example







**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')

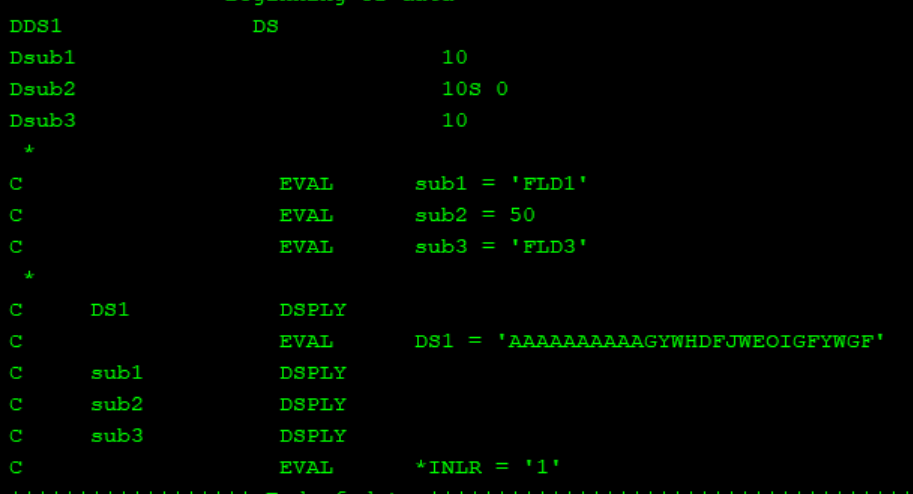
VI. PROGRAMME STATUS DATASTRUCTURE (SPECIFIED IN 'S')

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.

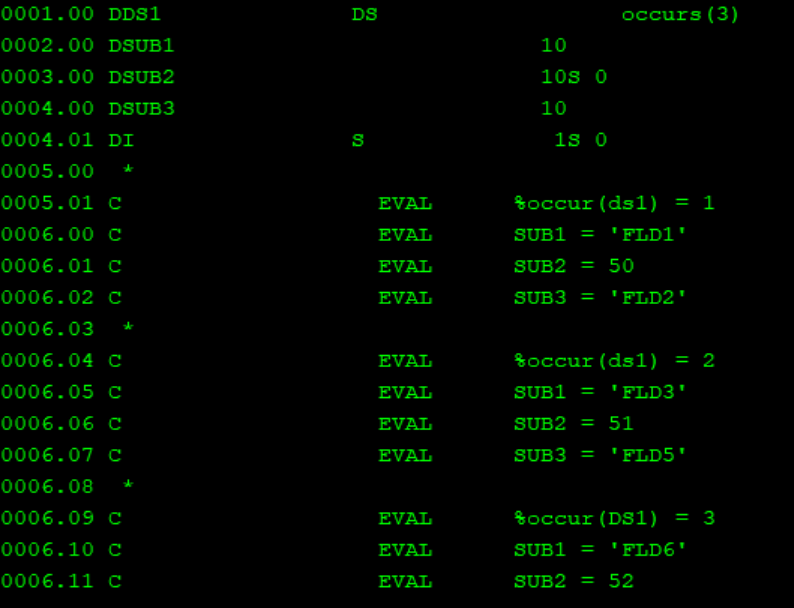
VII. FILE INFORMATION DATASTRUCTURE[minimum RRN/first RRN]

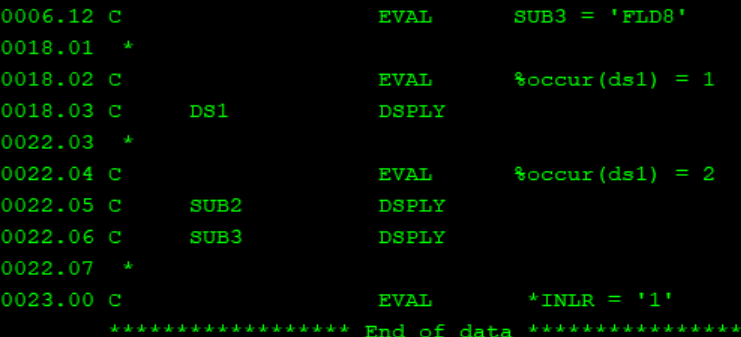
A file information data structure (INFDS) can be defined for each file to make file exception/error and file feedback information available to the program.

1. Simple data str (program described)

****

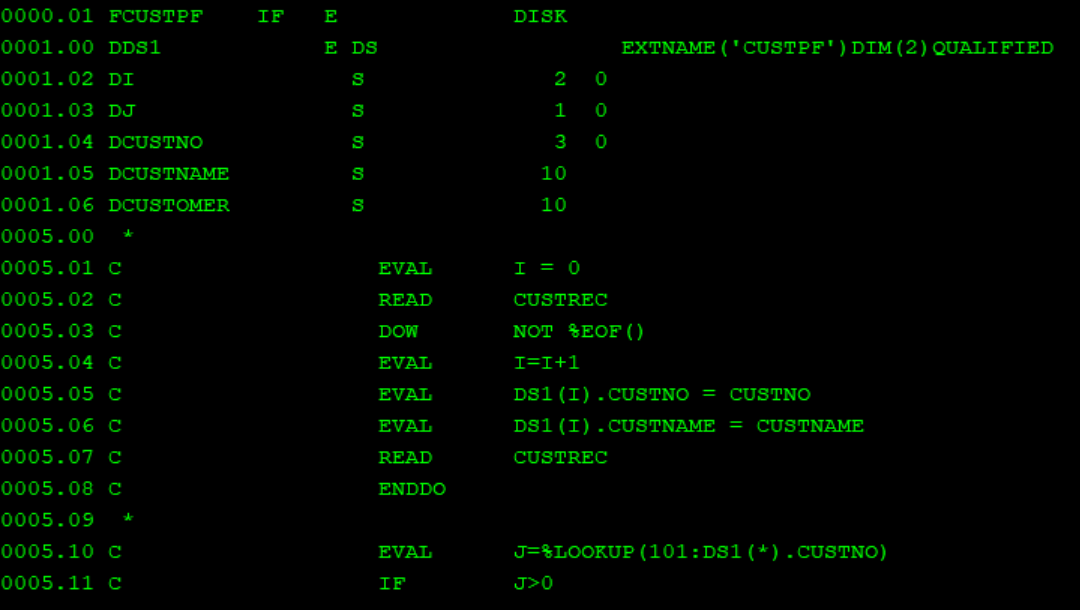
1. Multioccurance data structure

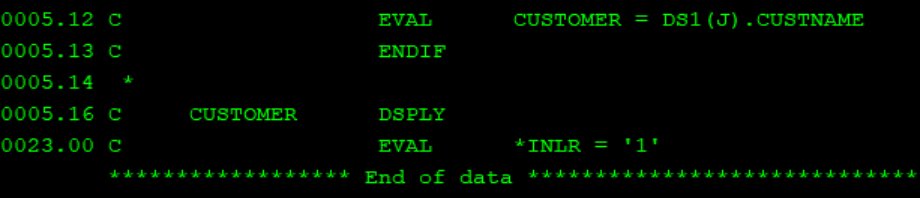
****

****

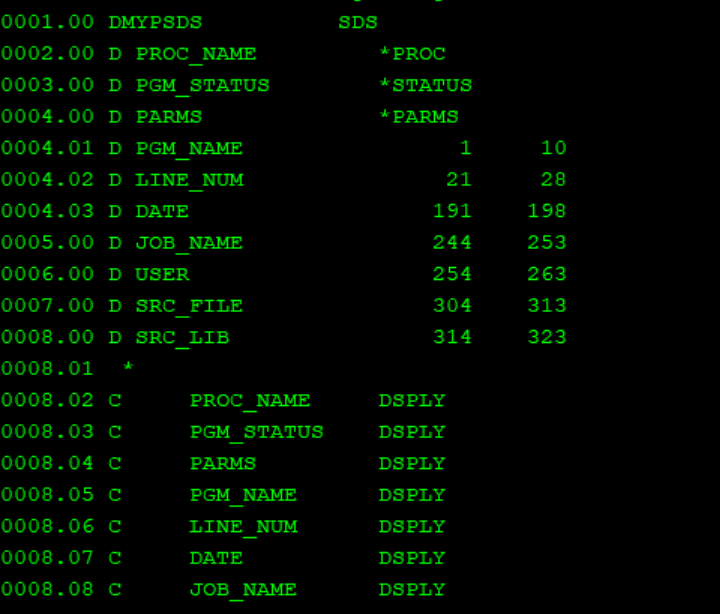
1. Externally described ds + array ds

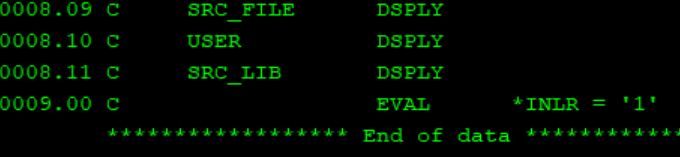
Here, using a given customer no. and array %lookup() customer name is found and printed.

****

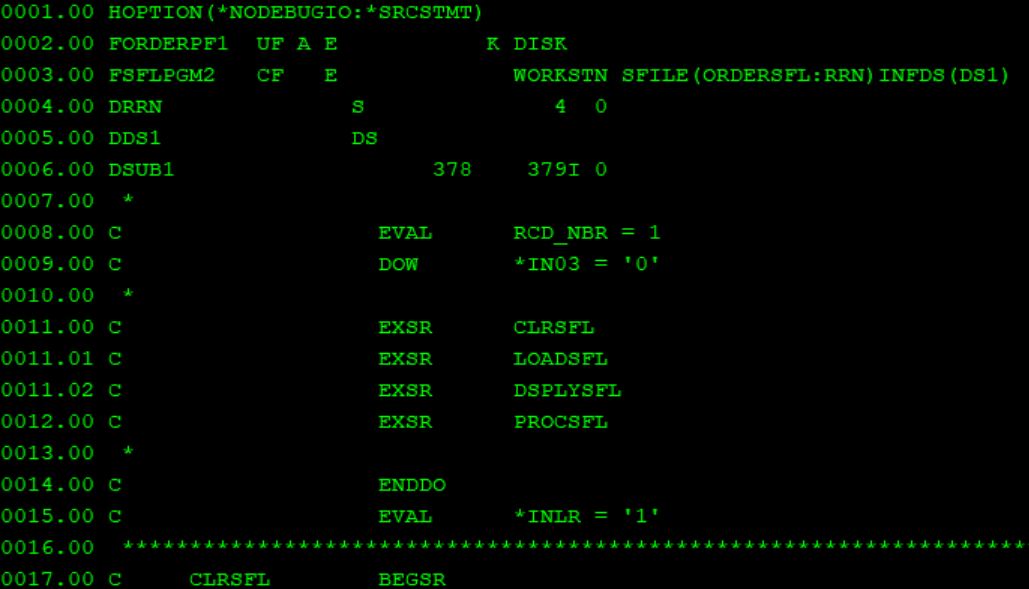
****

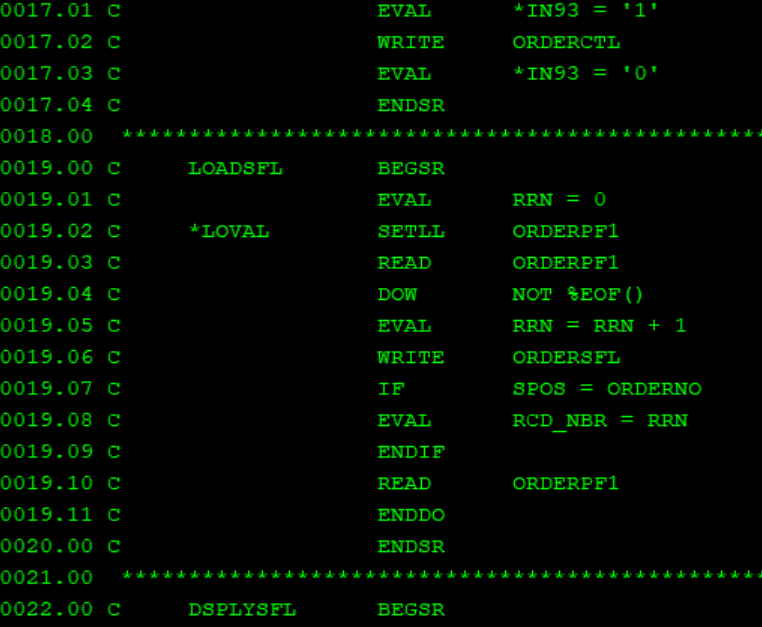
1. Program status ds(PSDS)

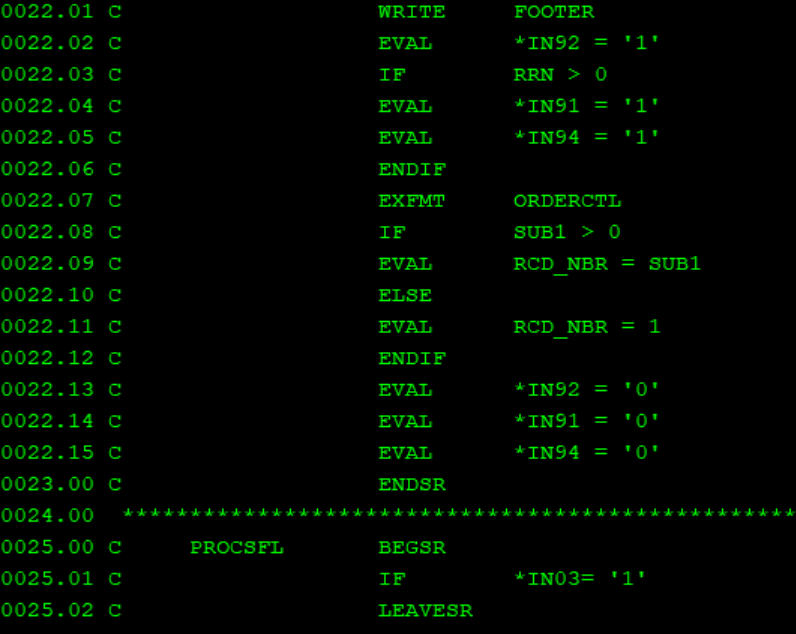
****

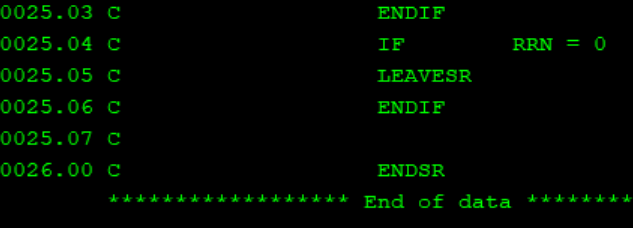
****

1. INFDS

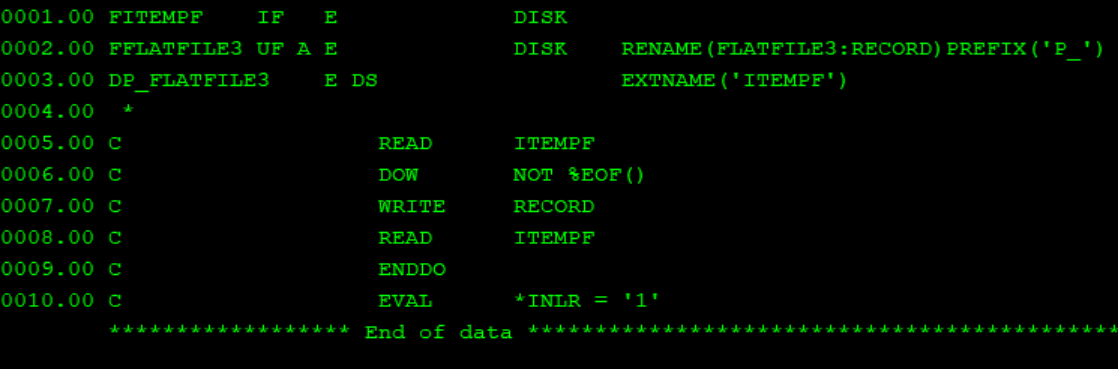
****

****

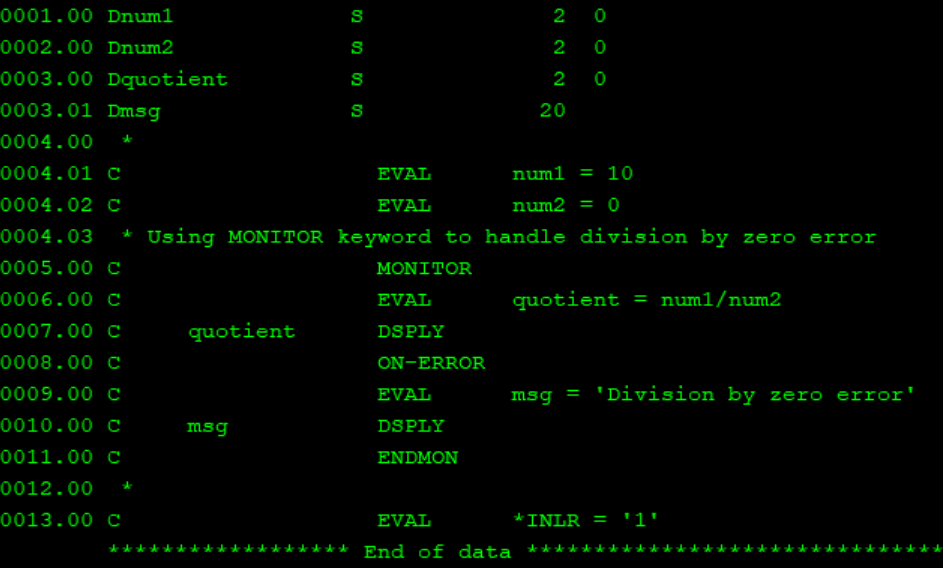
****

****

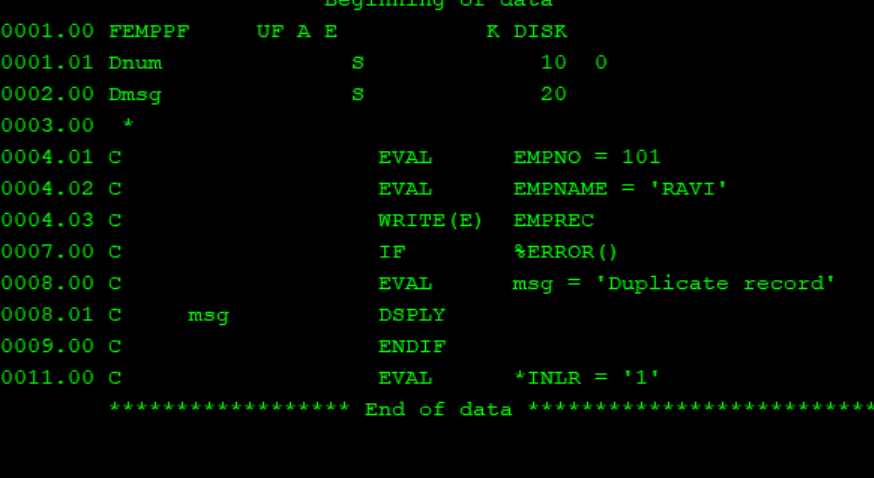
1. Reading from itempf and writing into flatfile

****

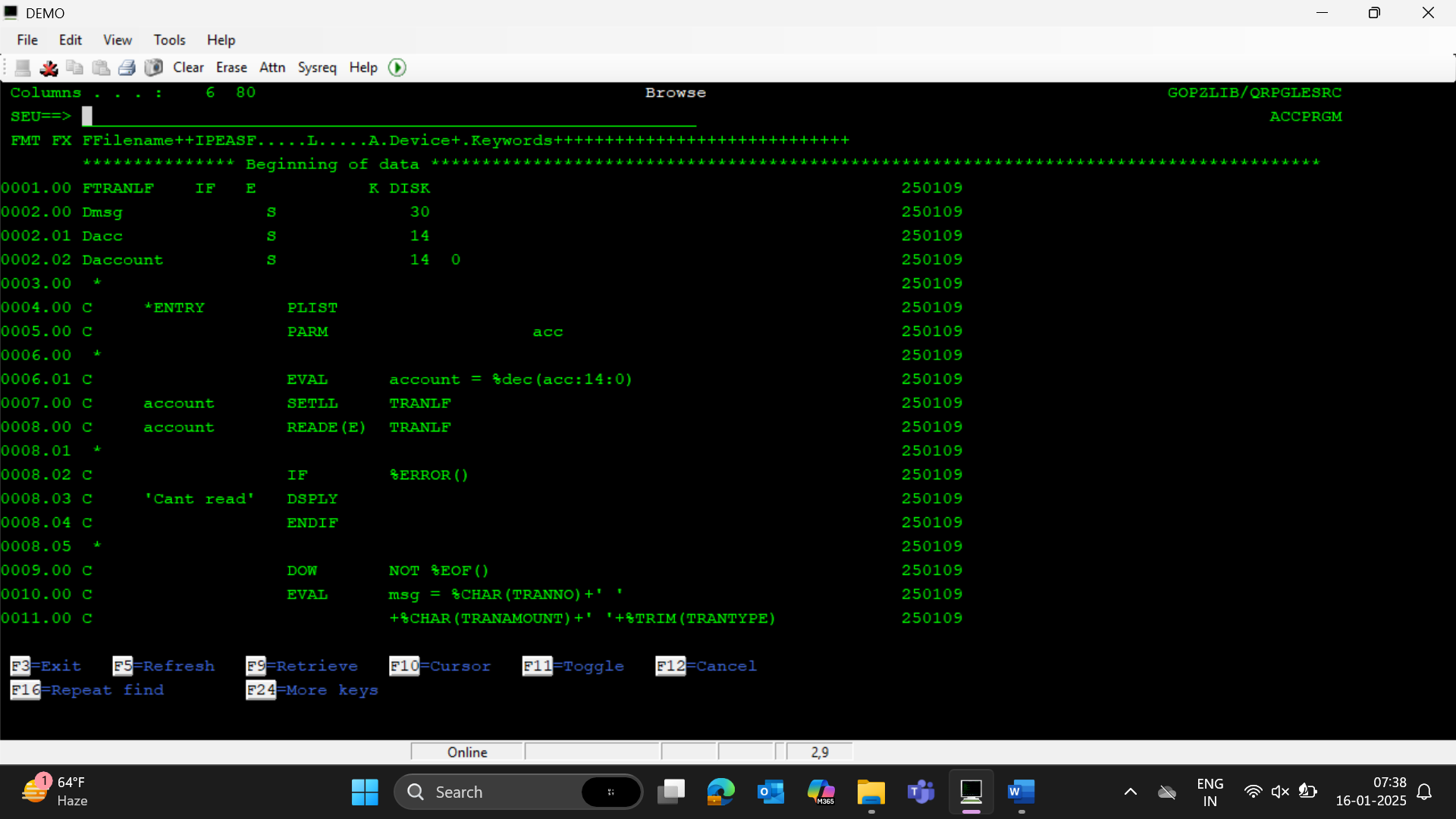
1. Block error handling

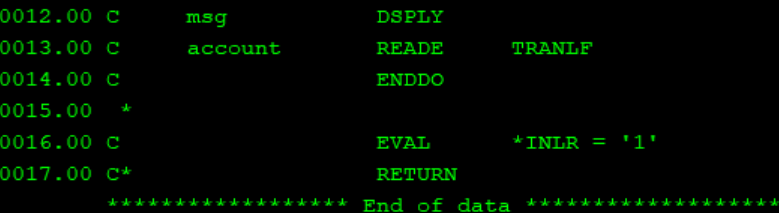
****

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

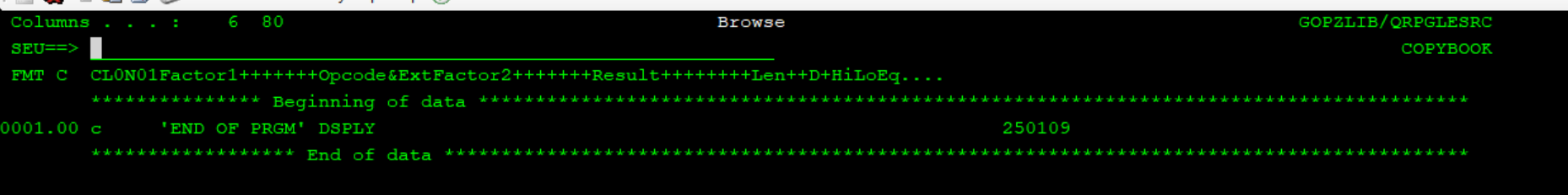
****

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

****

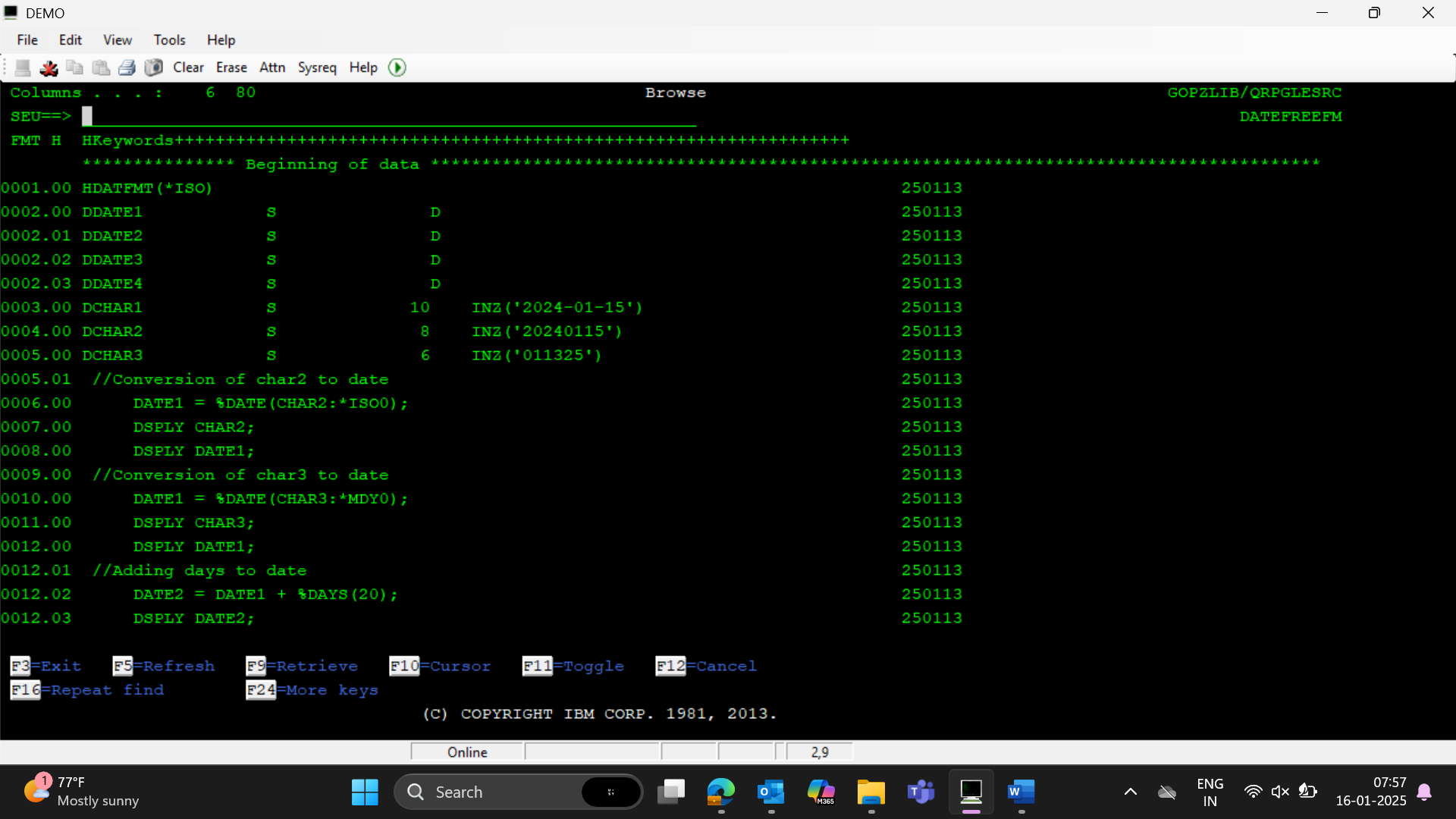
****

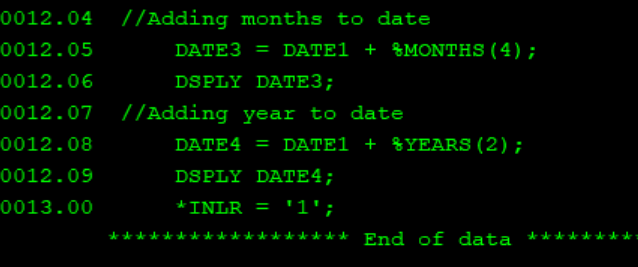
1. Copybook

****

****

1. Freeformat date

****

****