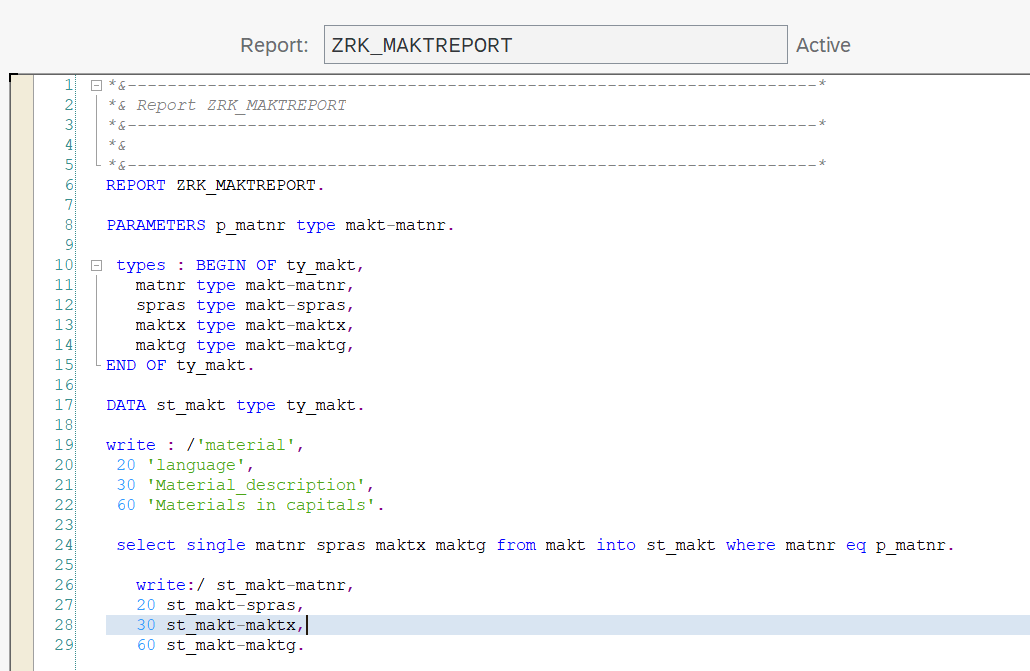
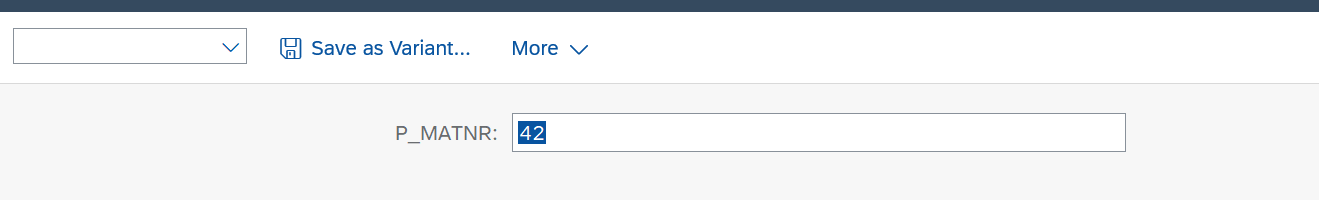
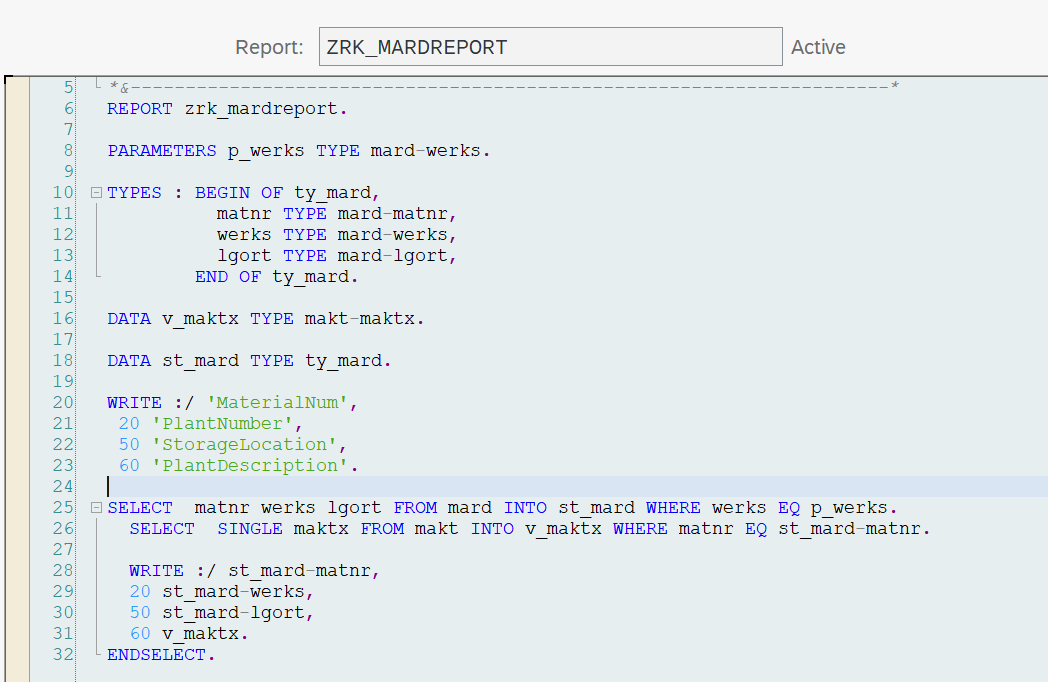
Assignment1: Displaying Material details by taking input material number through Parameter.



  
  
A screenshot of a computer

Description automatically generated

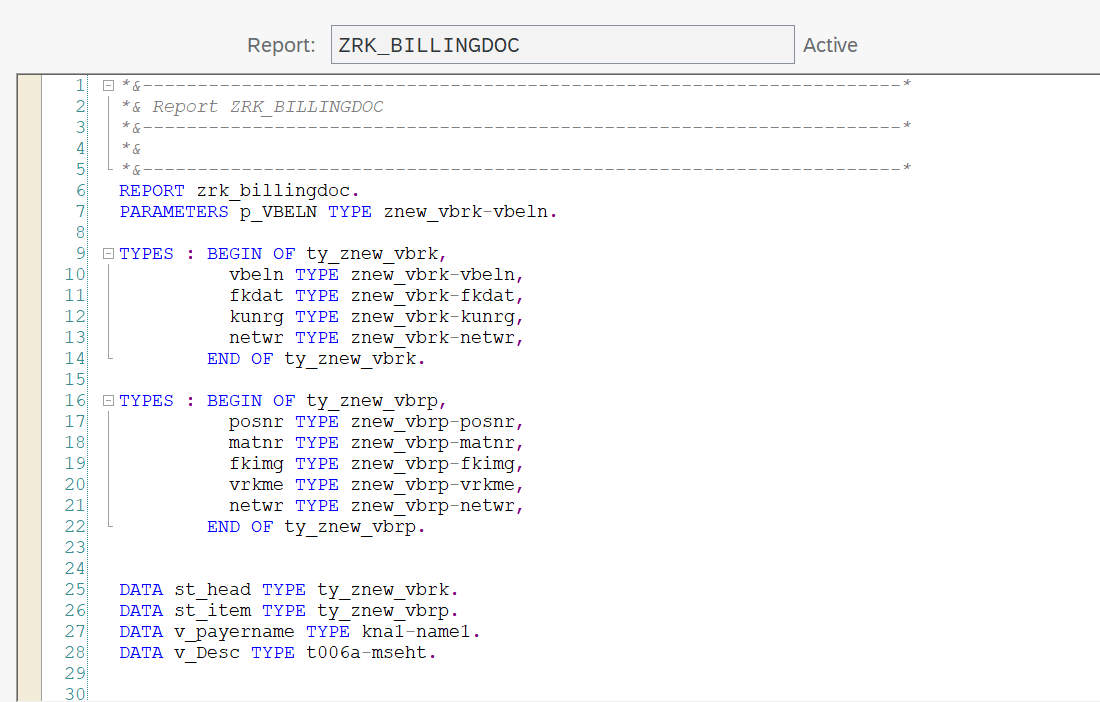
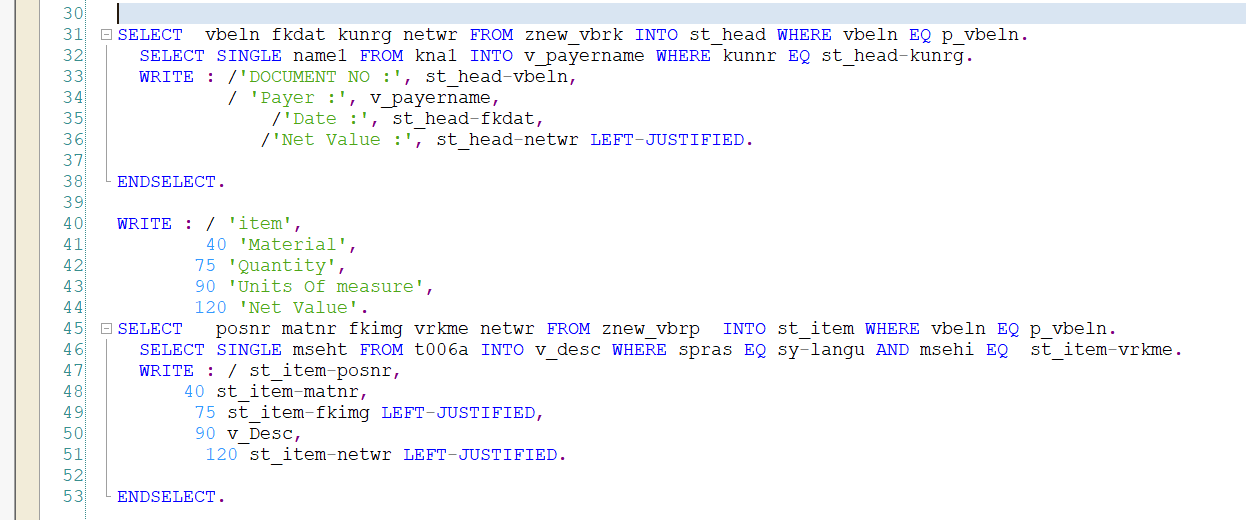
Assignment2:Program to show material description (which is in makt table) using plant number  


A screenshot of a computer

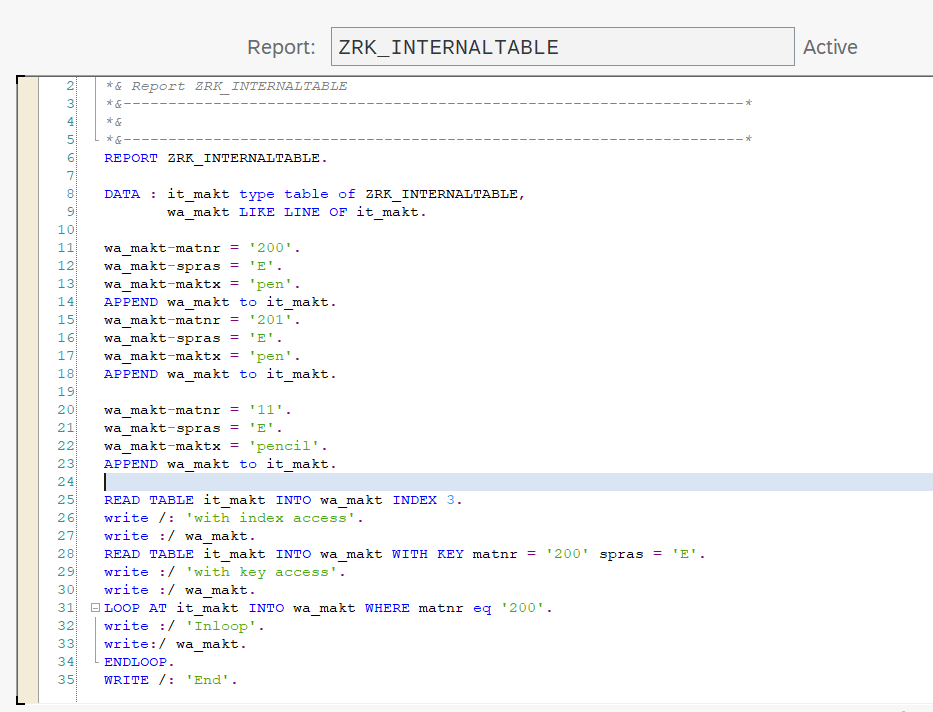
Description automatically generated

A screenshot of a computer

Description automatically generated

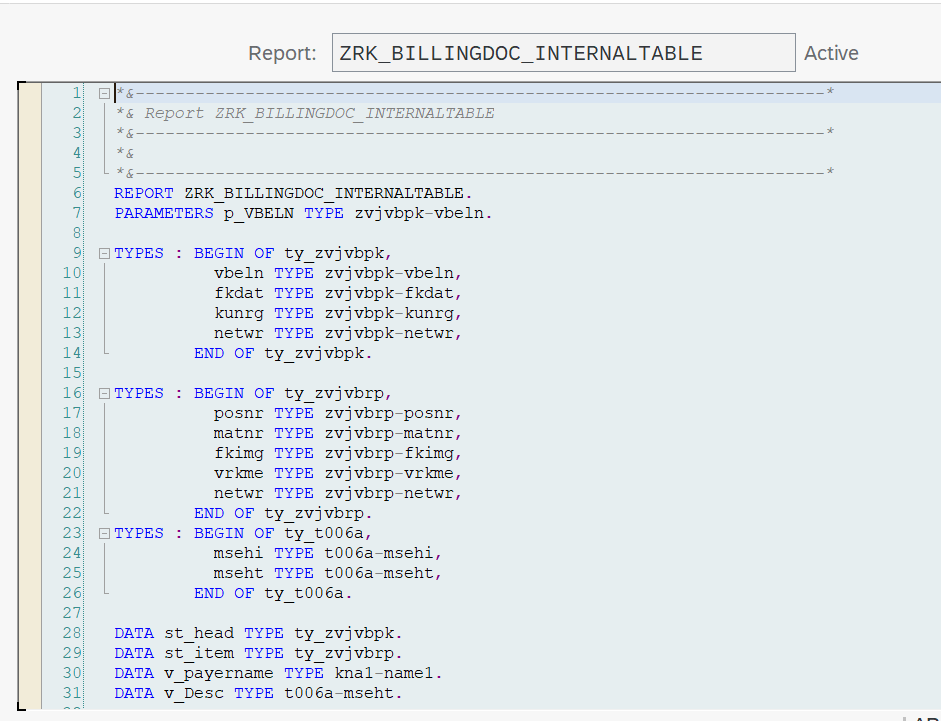
Assignment 3: display the dilling document details. Display header and item details header with payer name(which is in other table) item with unit of measure(which is in other table) used select ….ENDSELECT. and used only structure to display.  
  
  
  
A screenshot of a computer

Description automatically generated  
  
A close-up of a white background

Description automatically generated  
  
Assignment4: adding data to internal table and accessing data from it.  


A screenshot of a computer

Description automatically generated

Assignment 5: display the dilling document details. Display header and item details header with payer name(which is in other table) item with unit of measure(which is in other table) we used internal table and LOOP and ENDLOOP  
  


A screenshot of a computer

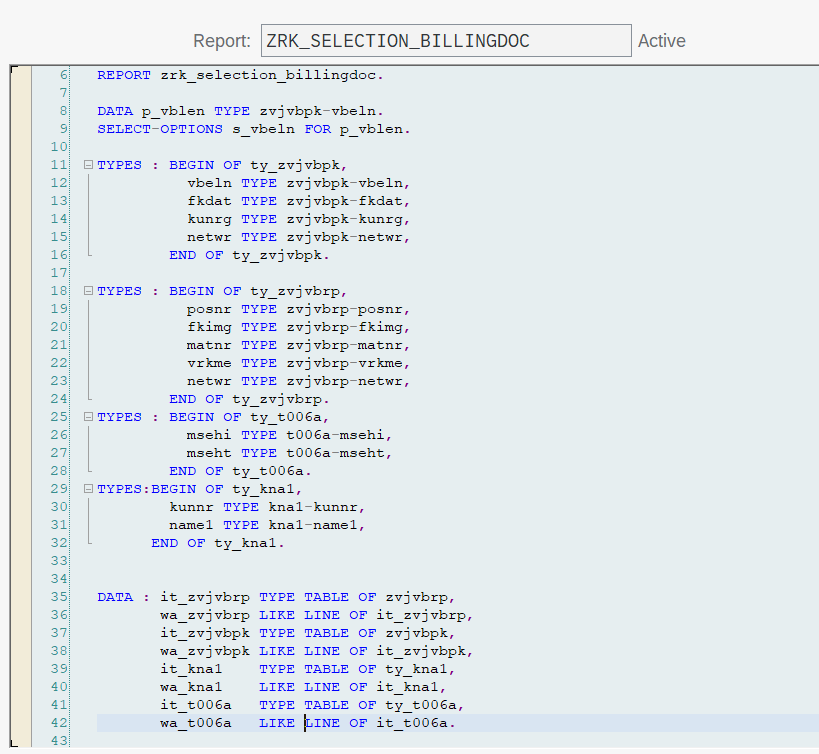
Description automatically generated

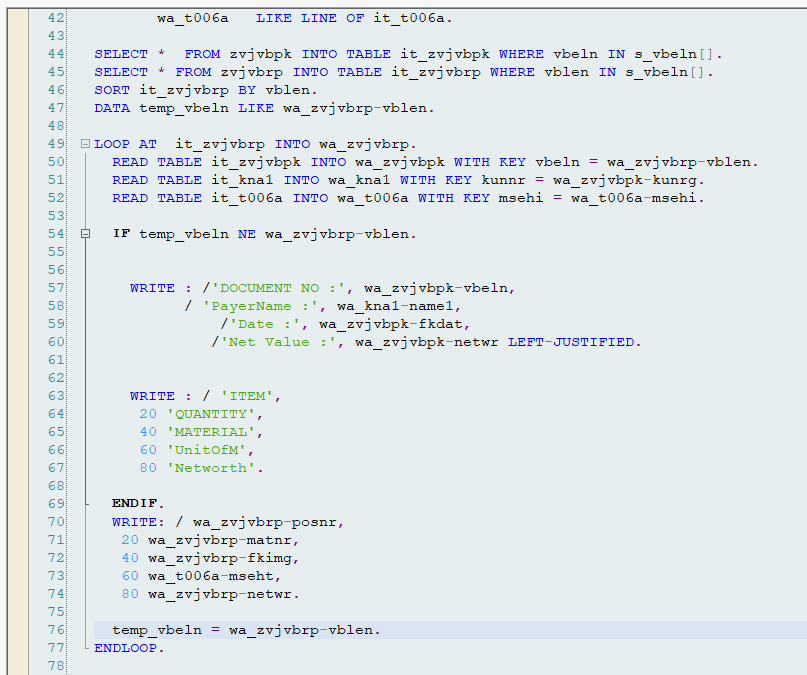
A screenshot of a computer

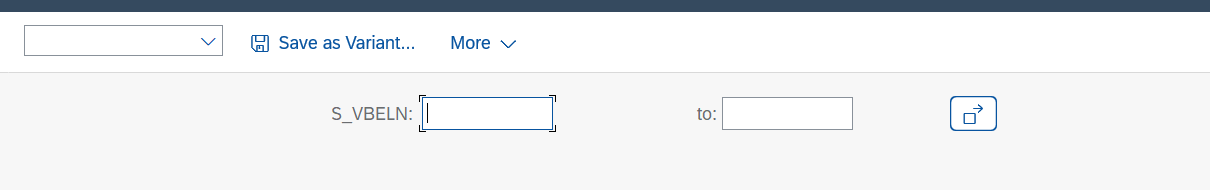
Description automatically generated

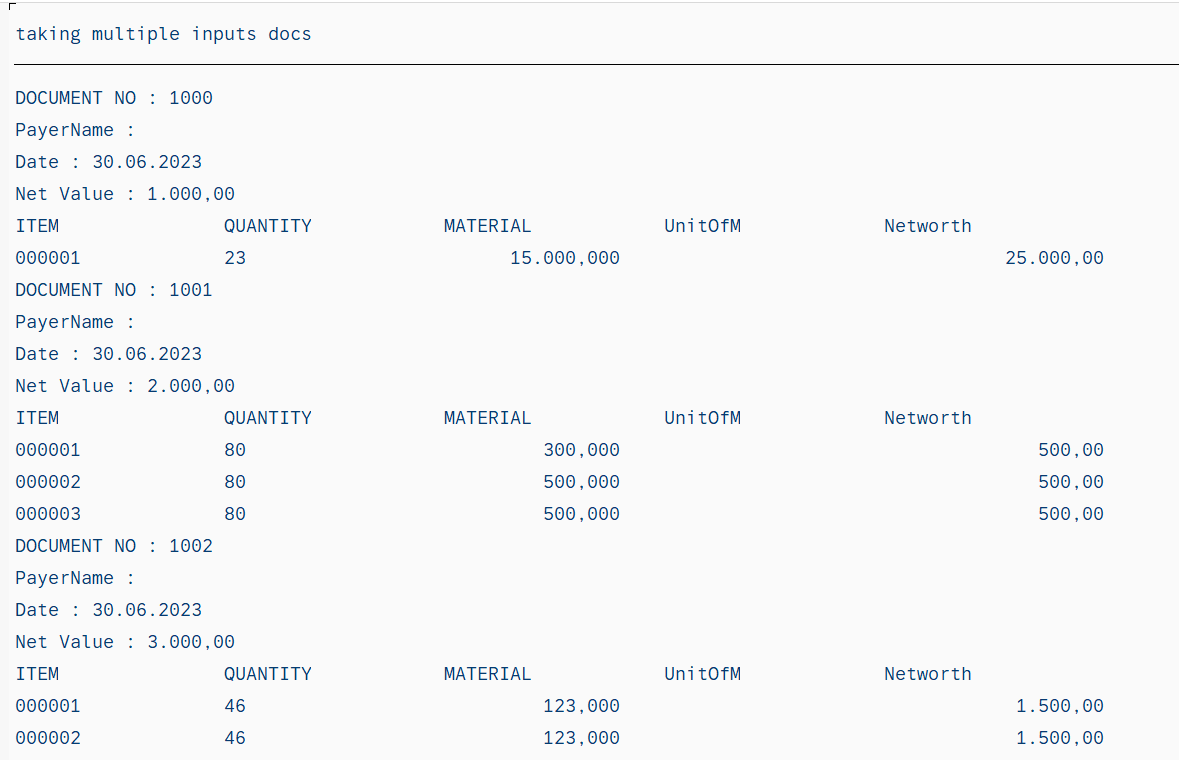
A close-up of a computer screen

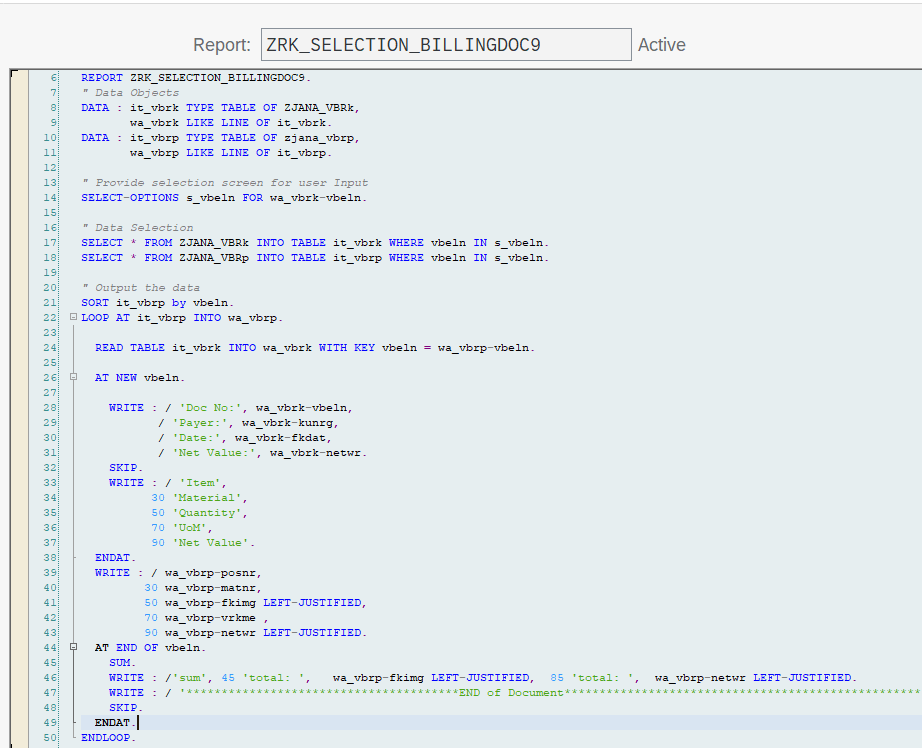
Description automatically generated

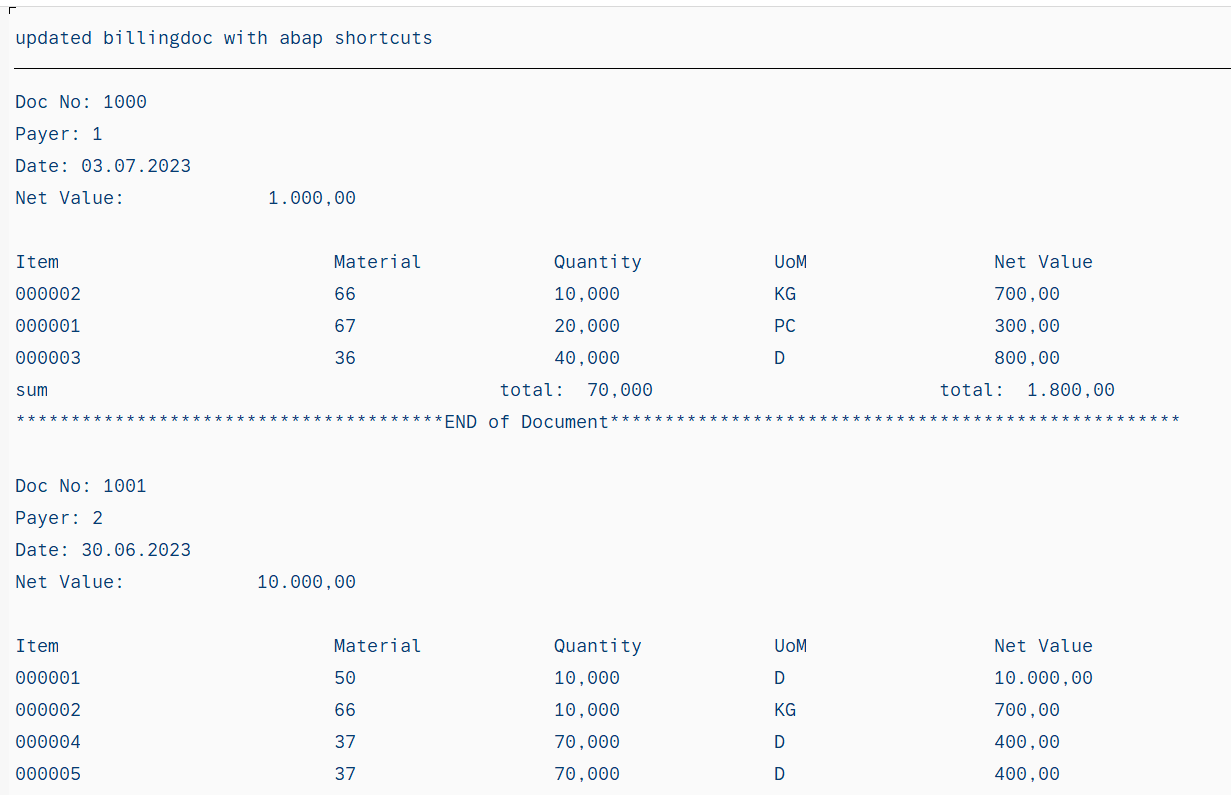
Assignmet6:Taking multiple inputs(document number) through select option. And displaying header and item data for given documents.  
  




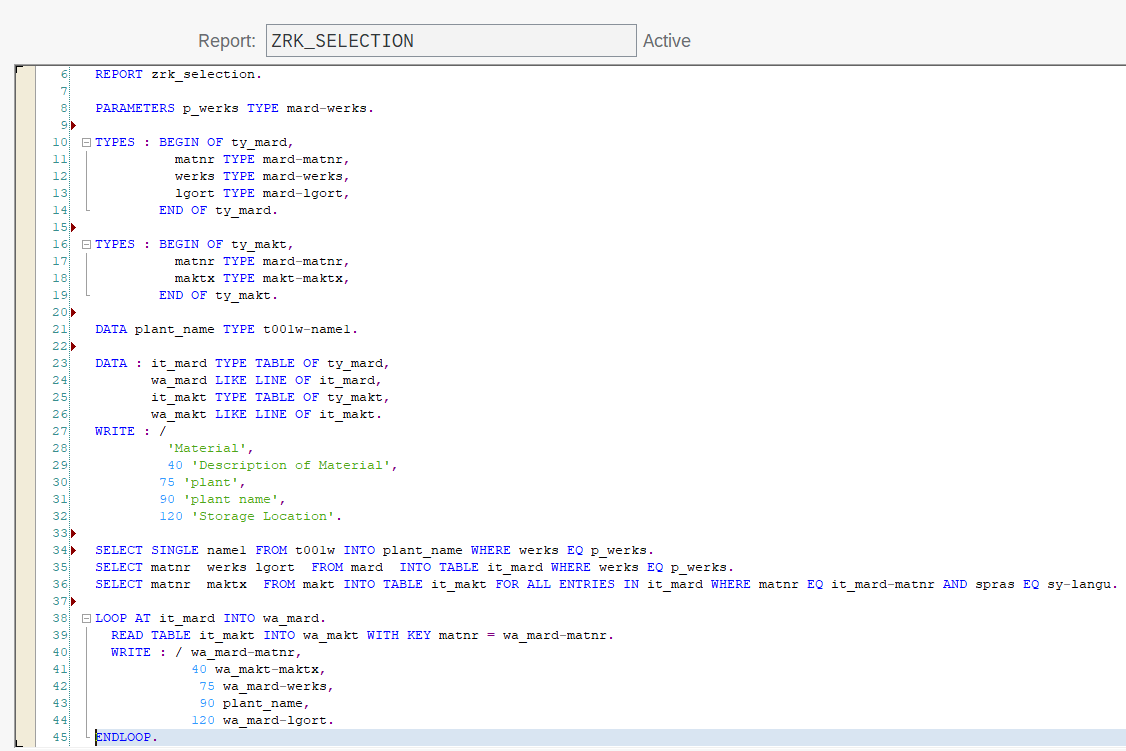


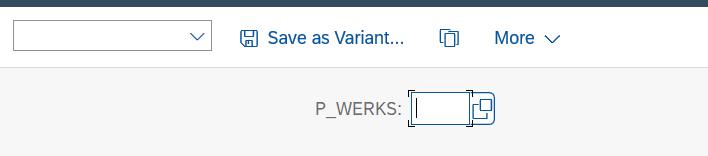


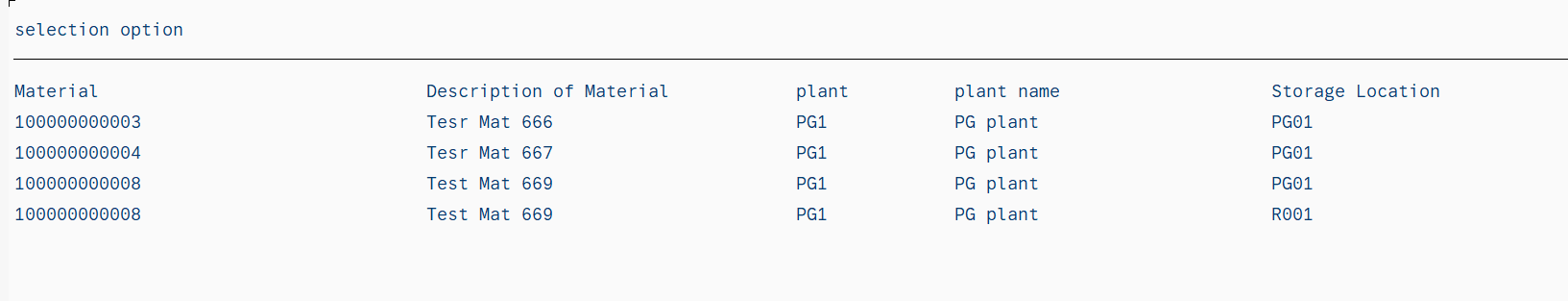
Assignment: 7 This below code includes 2 Rquirements. 1 is Printing sum of quantity and sum of net value.2nd is with Control statements.  




Assignment 8: Takes plant number as input and displays material, description of material, plant,storage location. Using parameter.

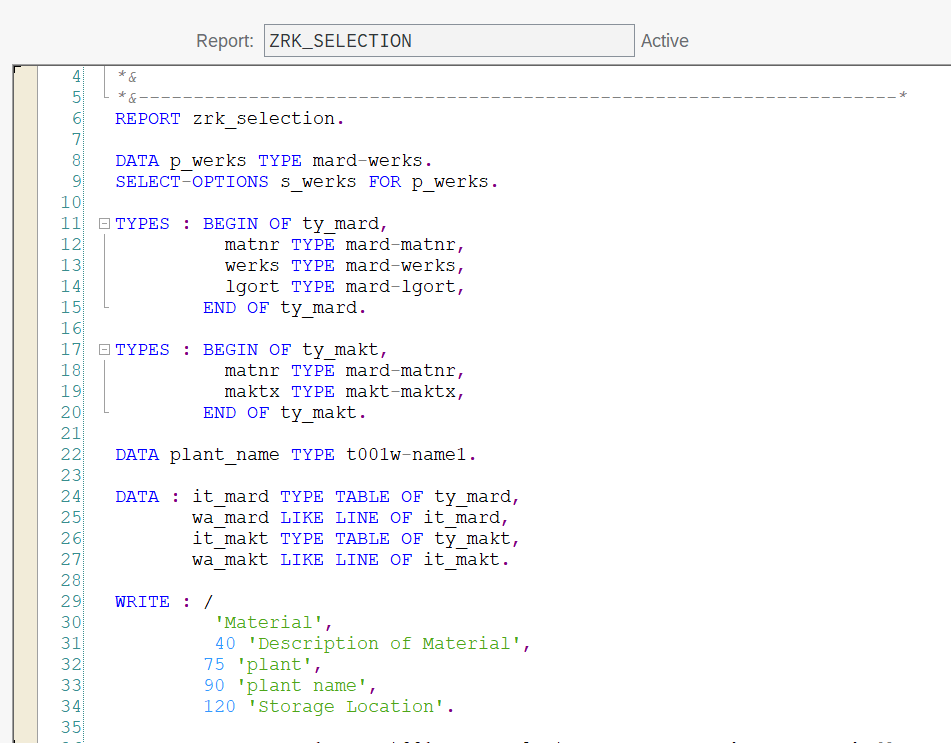


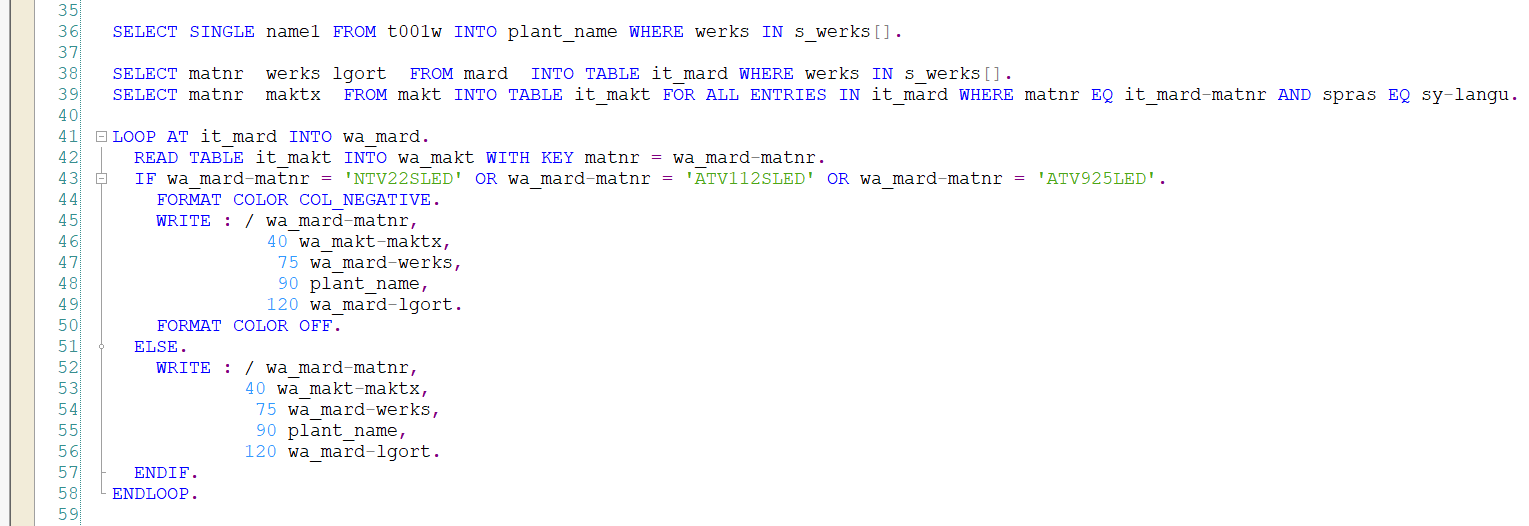


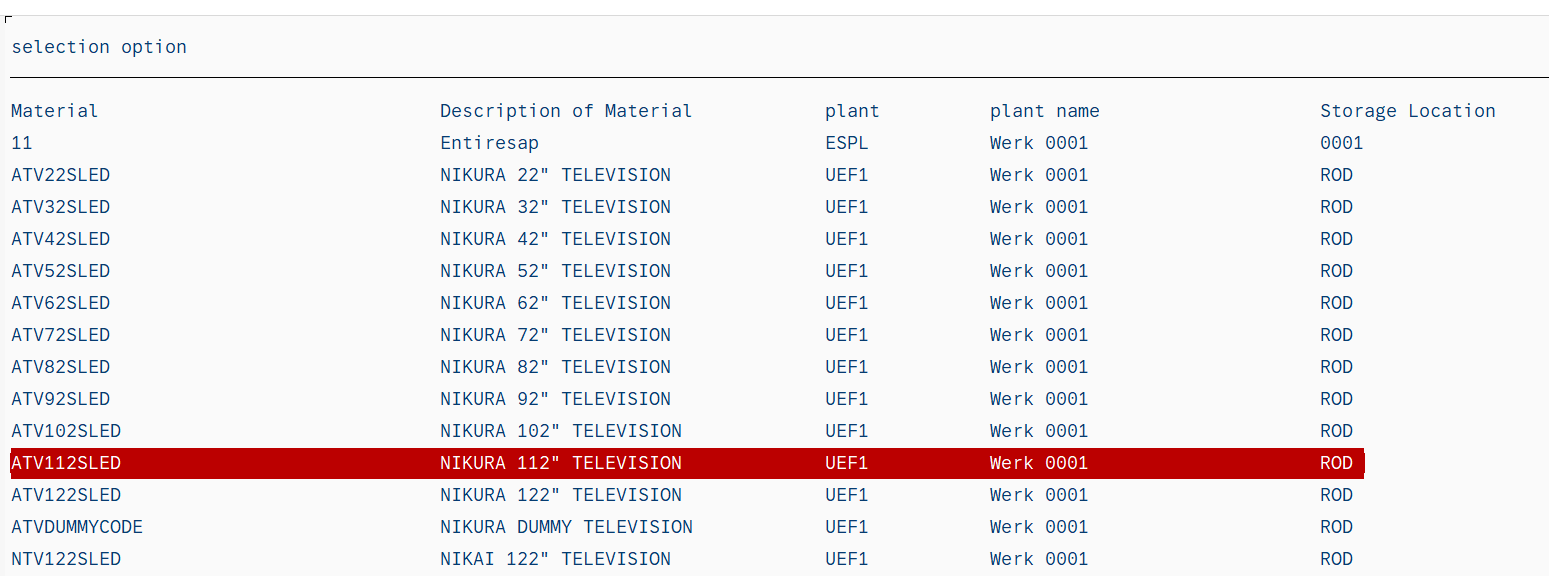


Assignment 9: Below program takes input Plant numbers and prints material, material description, plant ,plantname, storage location,. And for given 3 dangerous material(ATV112SLED,NTV22SLED, ATV925SLED) it will print in red colour. Using select-option

CONCEPTS: select-options, structure type,internal table, work area table, read internal table using READ, LOOP.



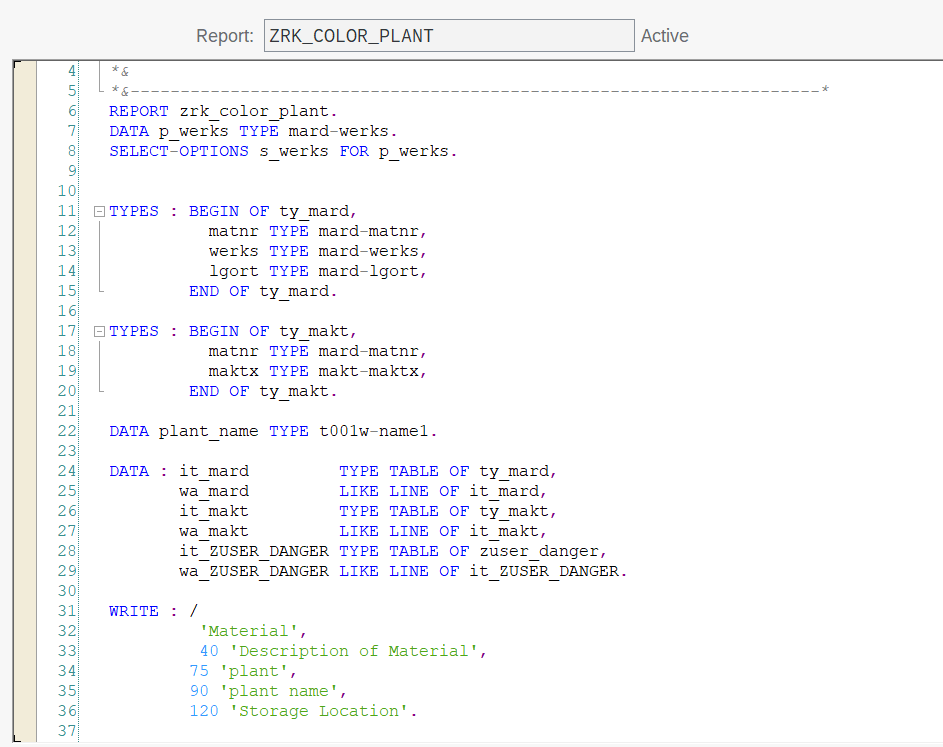


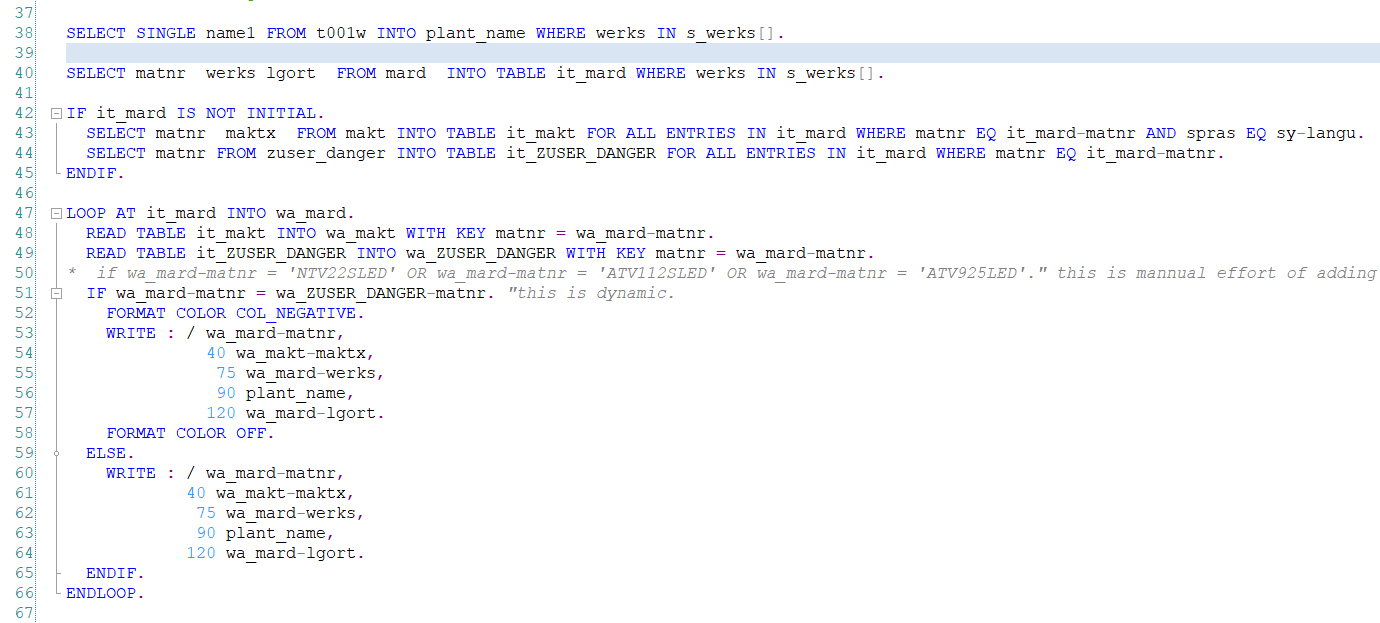


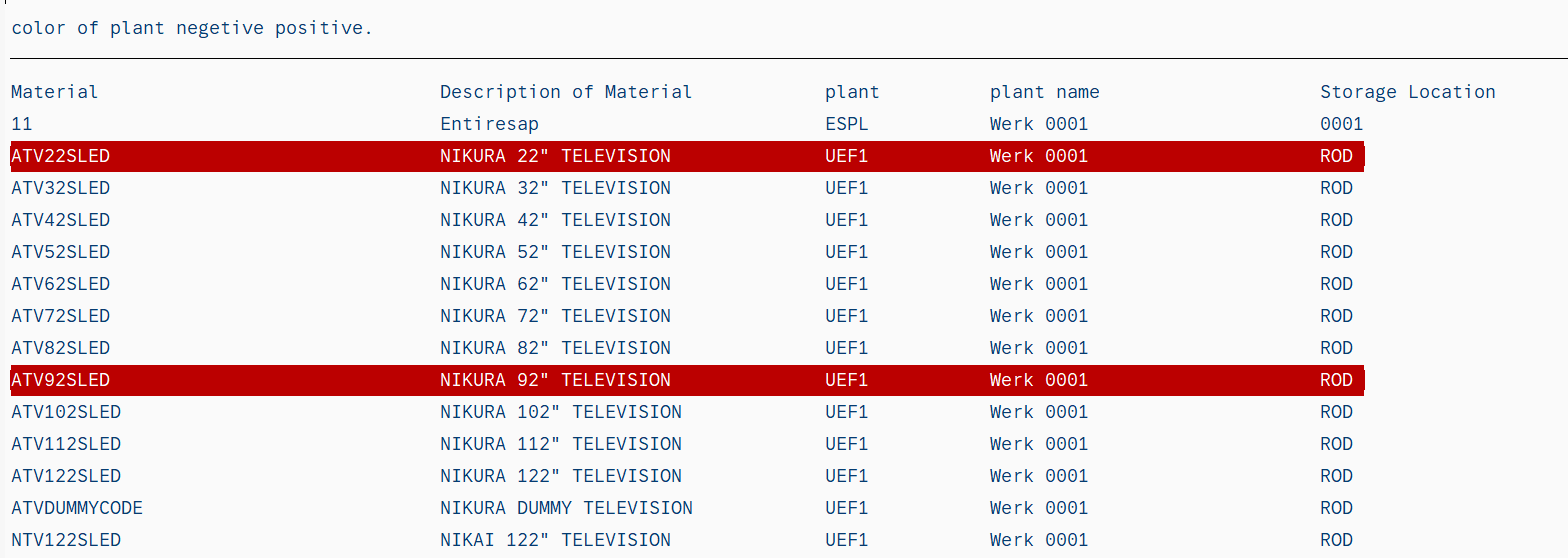
Assignment 10: IN ABOVE PROGRAM WE DEFINED DANGEROUS MATERIAL NUMBER IN CODE ITSELF

SO WE ARE MAKING IT DYNAMICALLY. we kept those dangerous material in separate table.

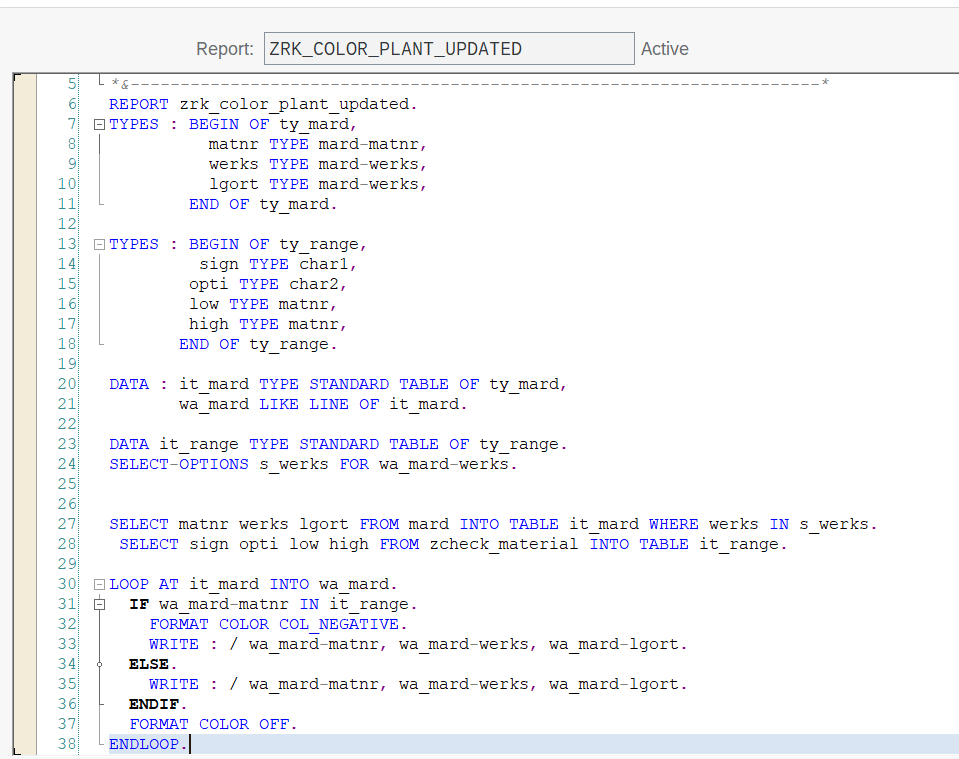
And compared.  And made color red for those details.

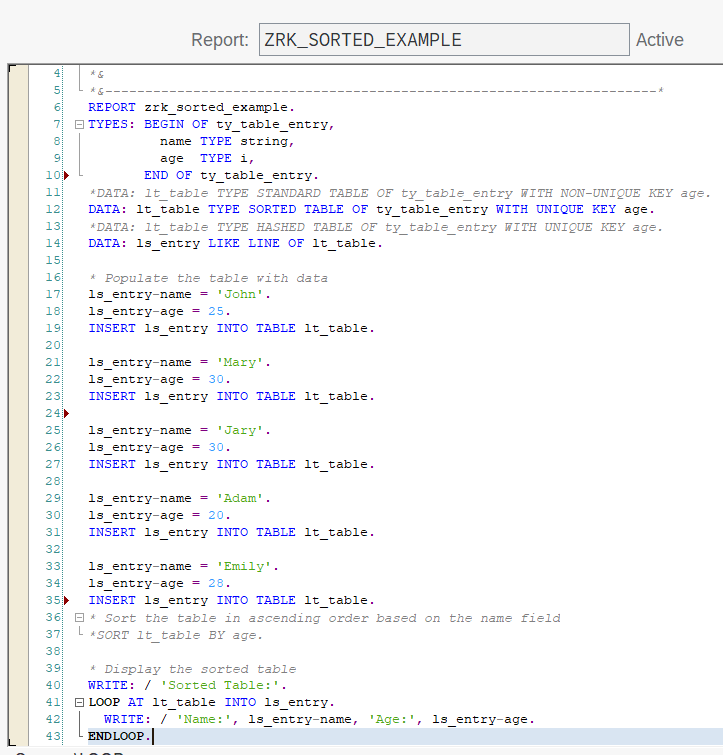


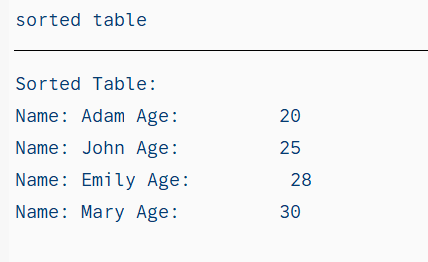


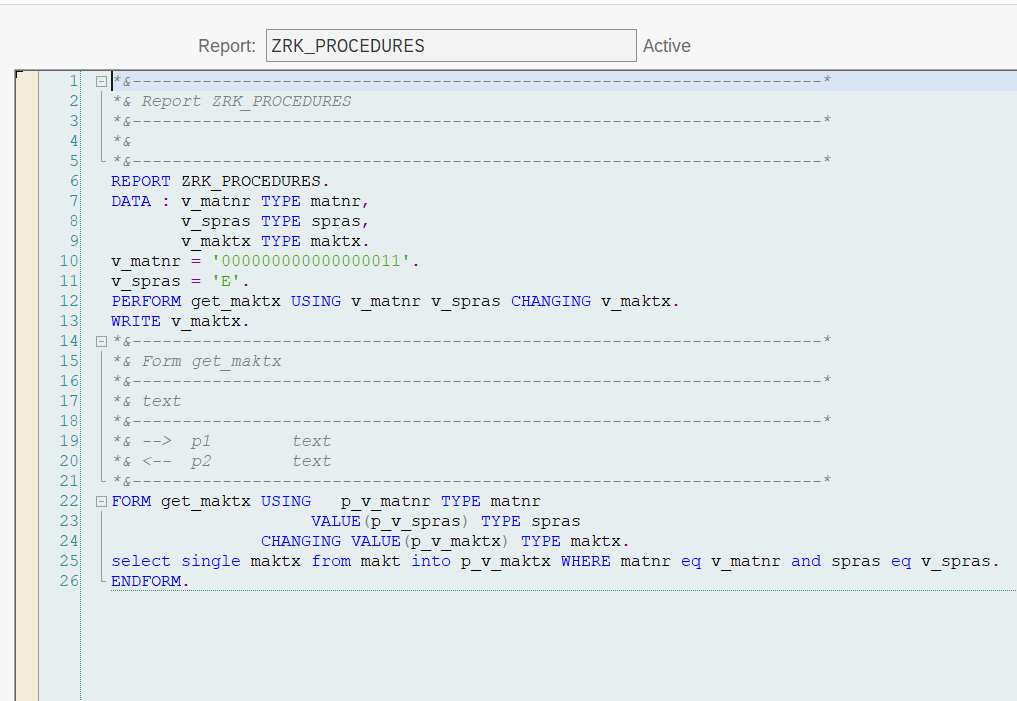


Assignment 11:  
Re modifying above code or more accurate code: my idea user can select exclude include through select-option and save the variant. Next time If he wants to add he can add in the variant.



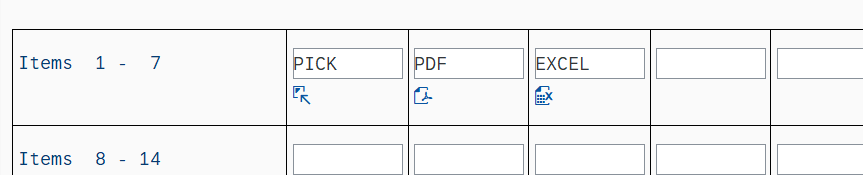
Assignment12:This code includes adding data to different types of internal table.Standard, sorted, hashed.  
  


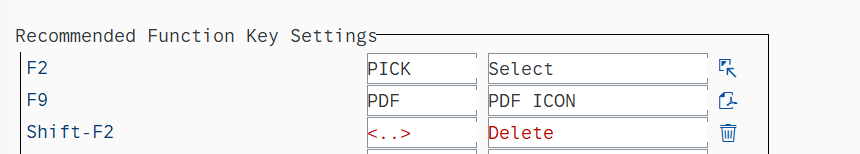


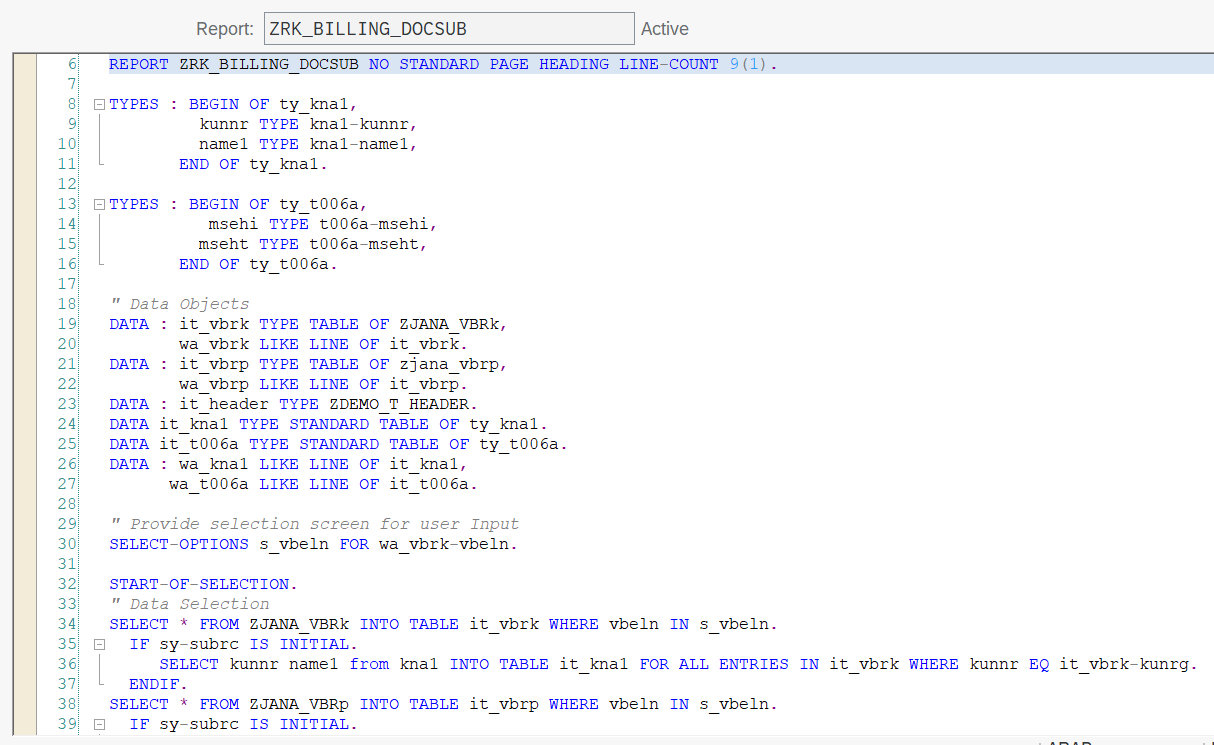
Assignment13: Subroutin created and called in the same program.  


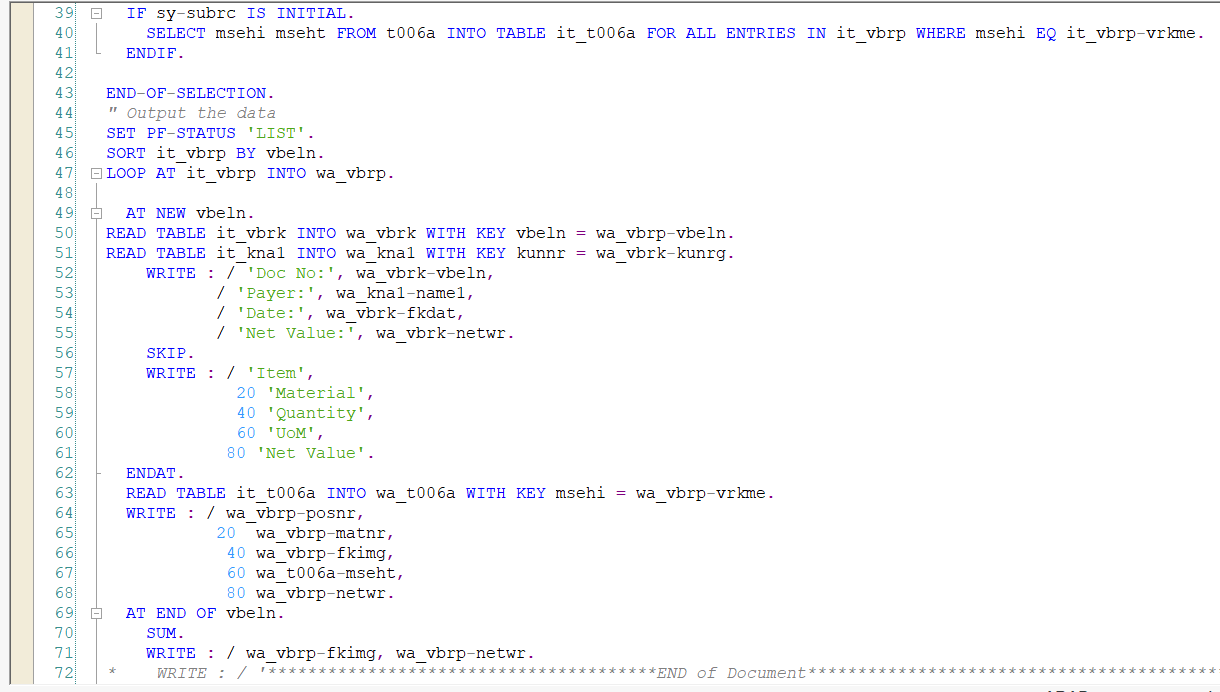


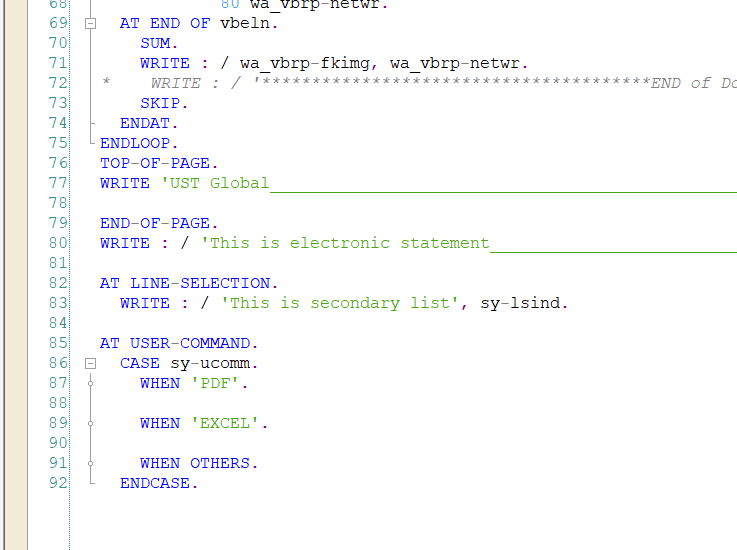
………………………………………………………………………………………………

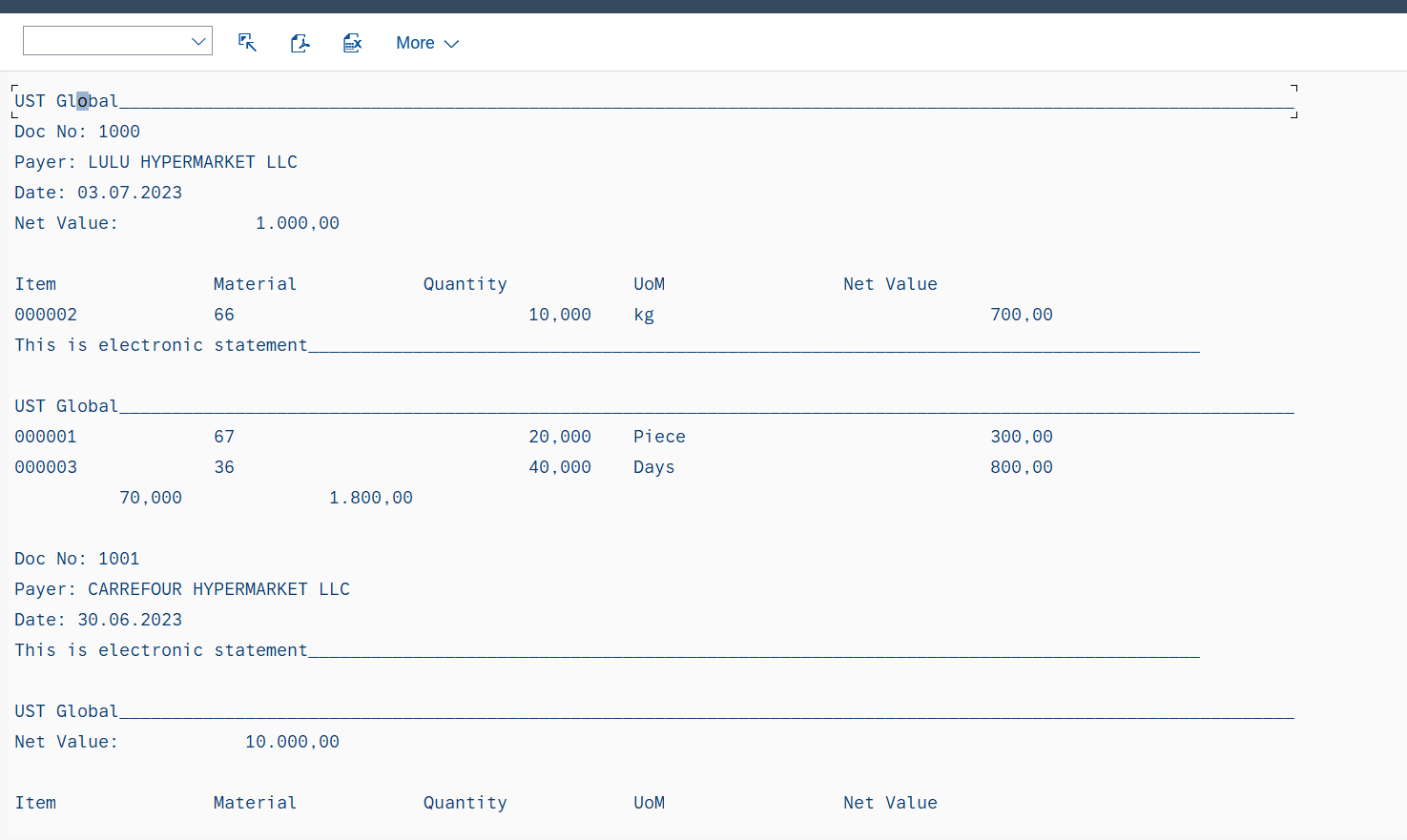
Assignment 14 : Code Modularization, used Start-of-selection, End-of-selection, At user-command, At line-selection, Top-of-page, End-of-page, PF-status.  


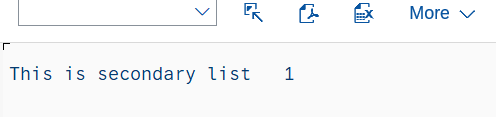






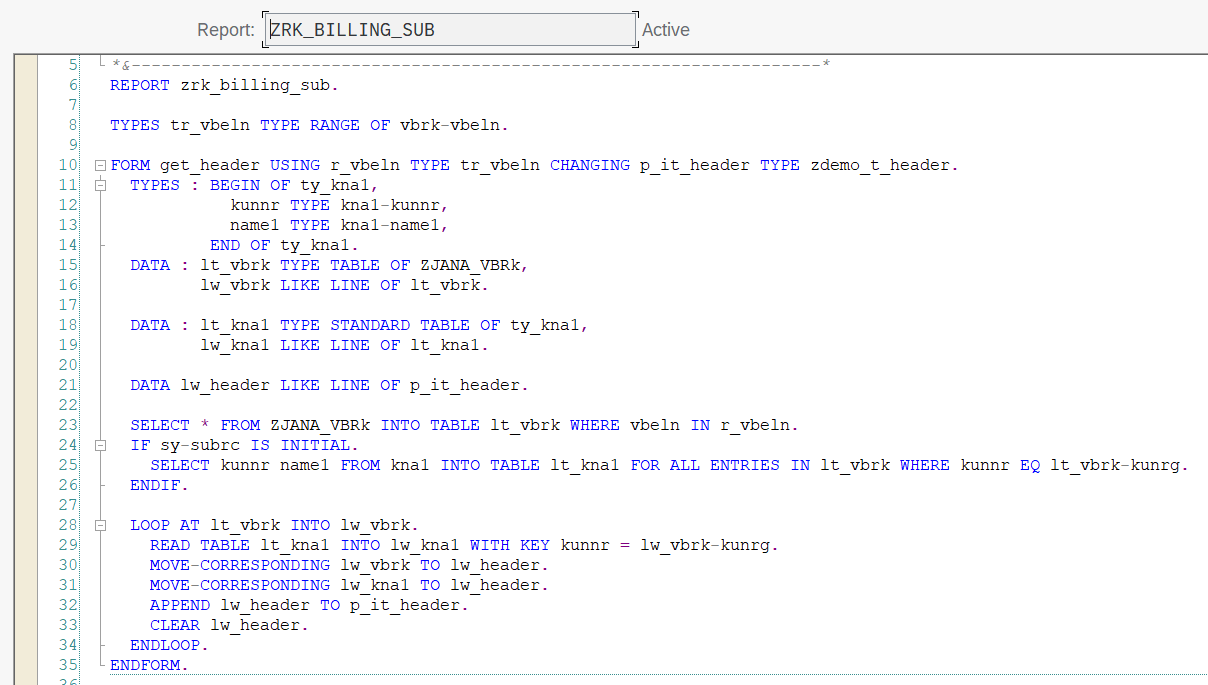


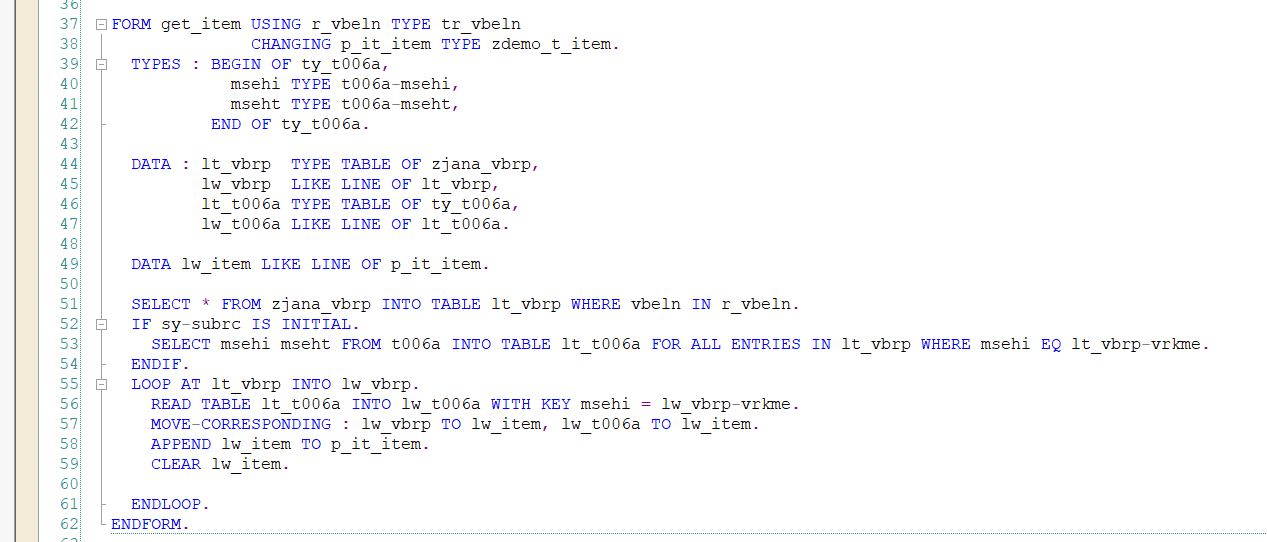


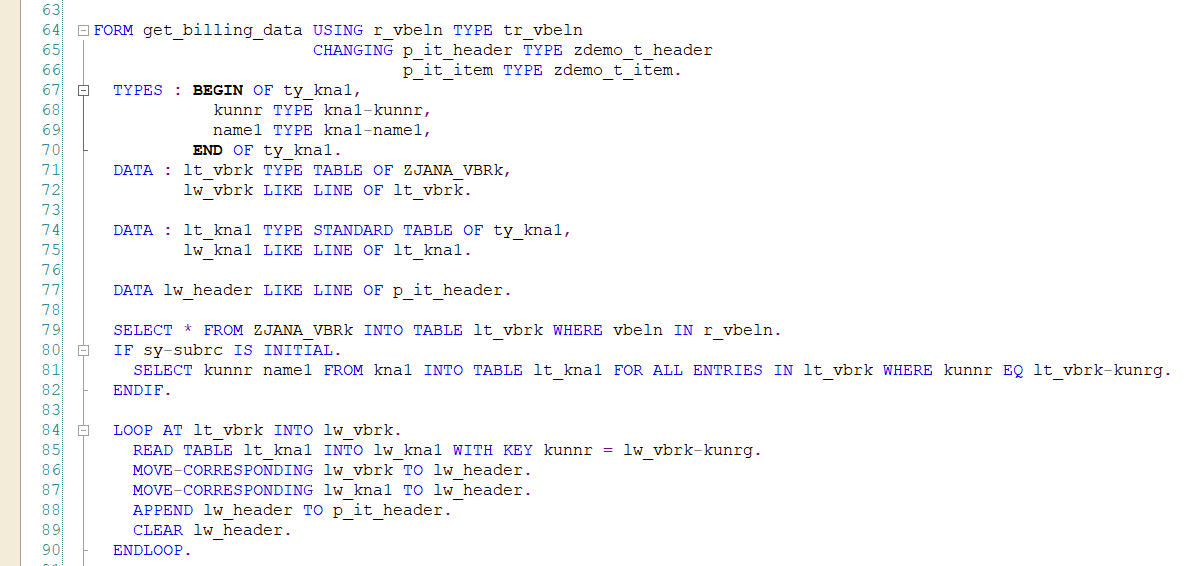


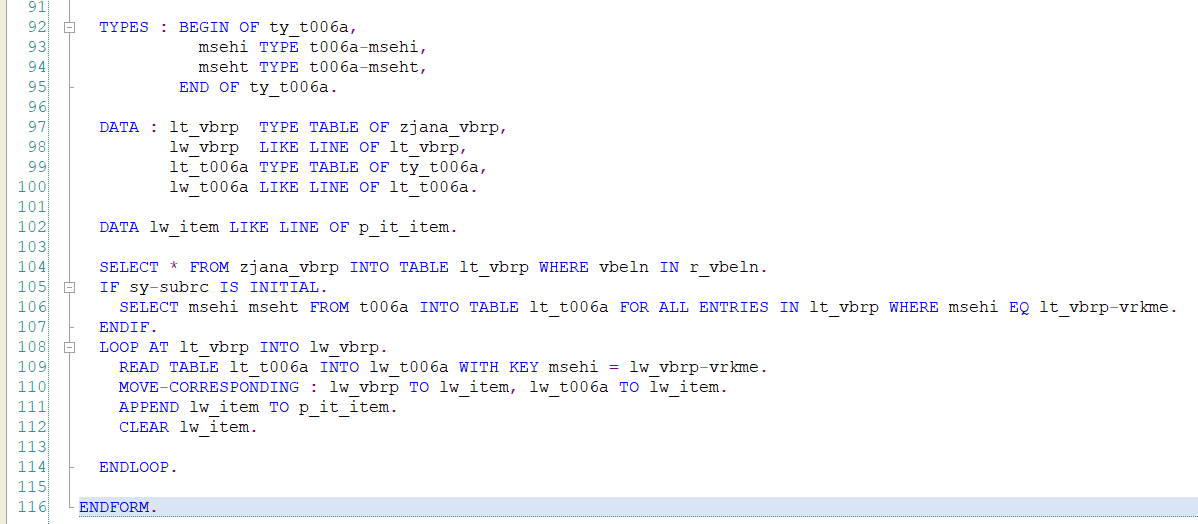
…………………………………………………………………………………………………

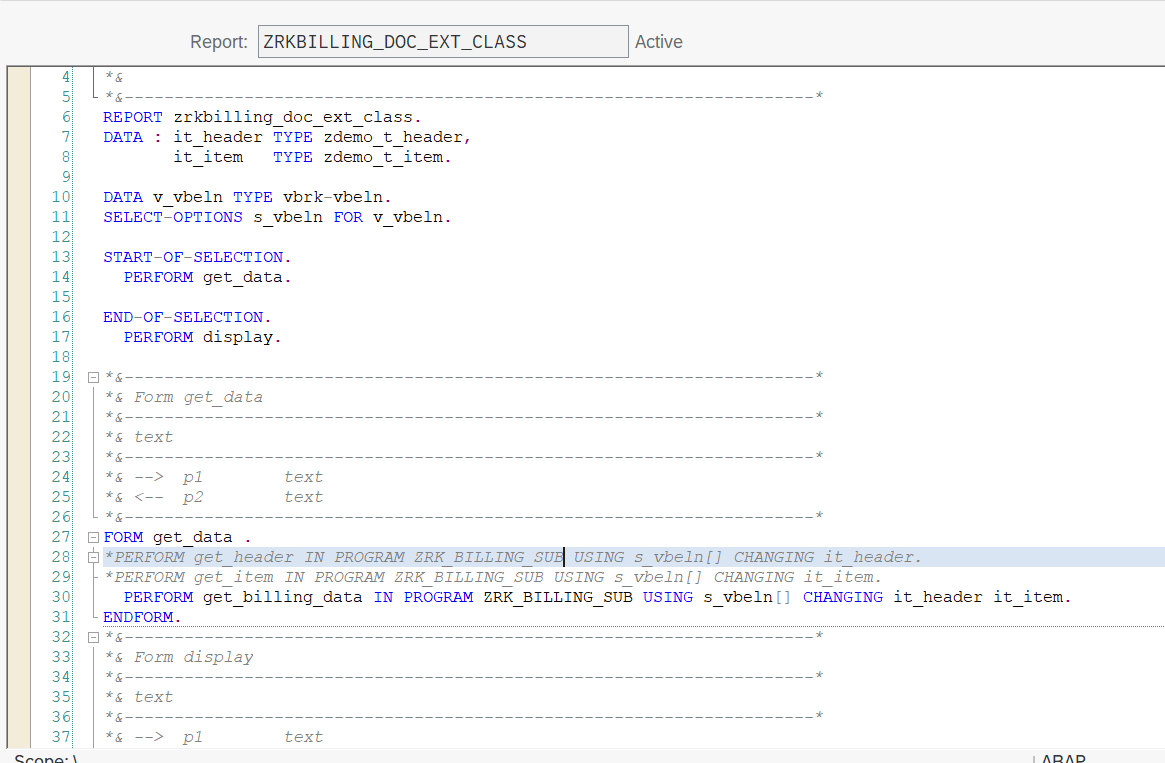
Assignment 15:  
In below code subroutine created in one program and called in another program.  
Below are subroutines, In these one is for getting header data,other one is getting item data, one more subroutine includes getting header and item data.

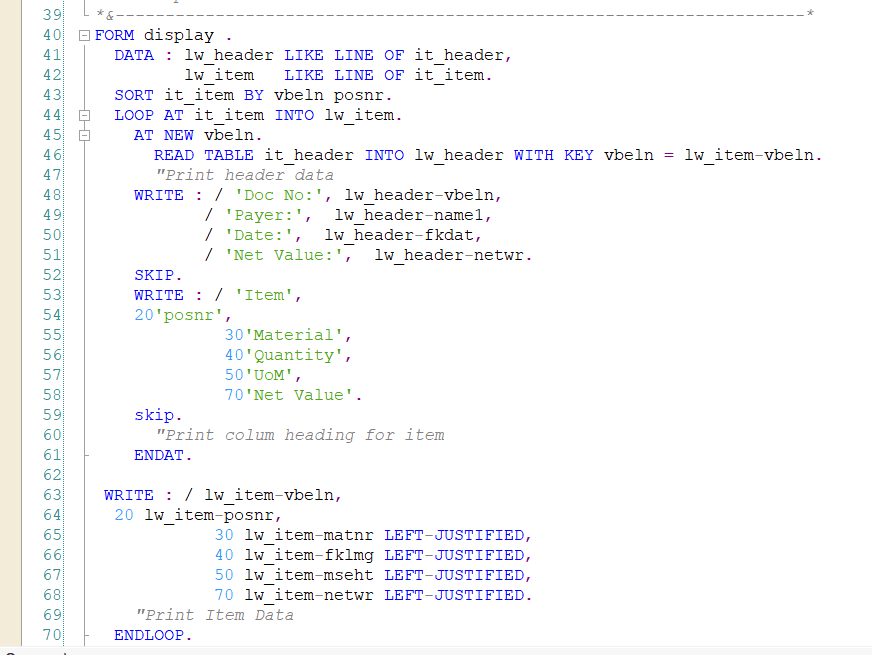
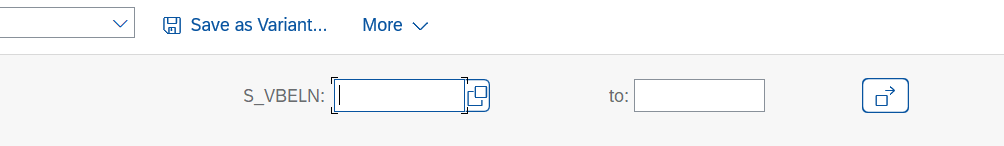


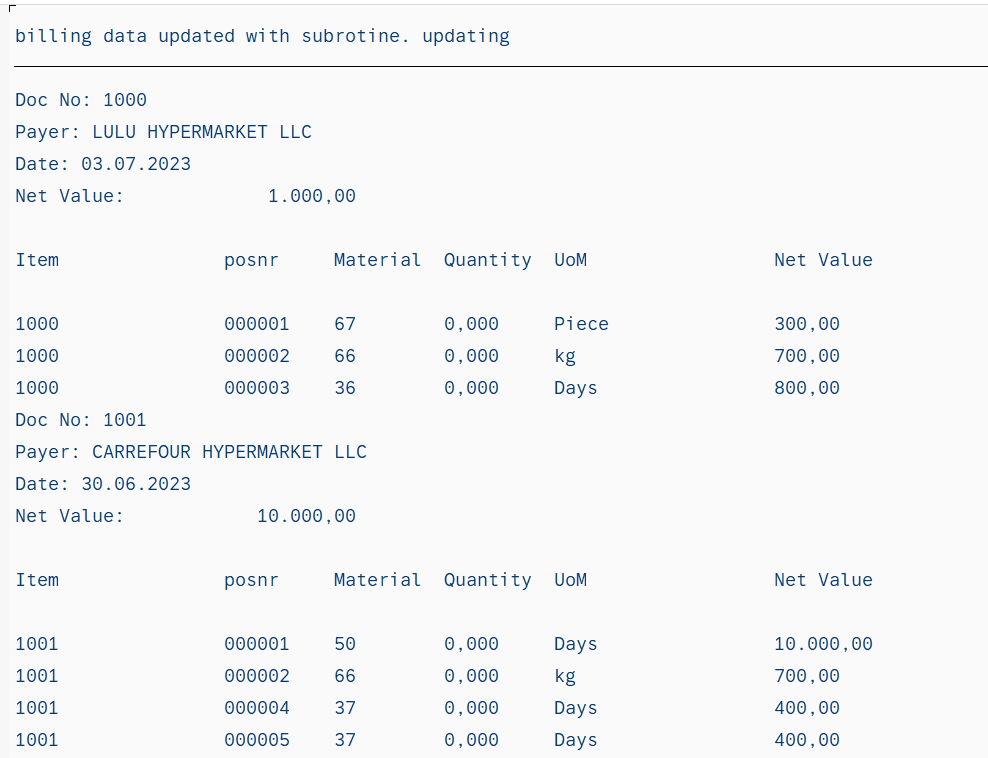




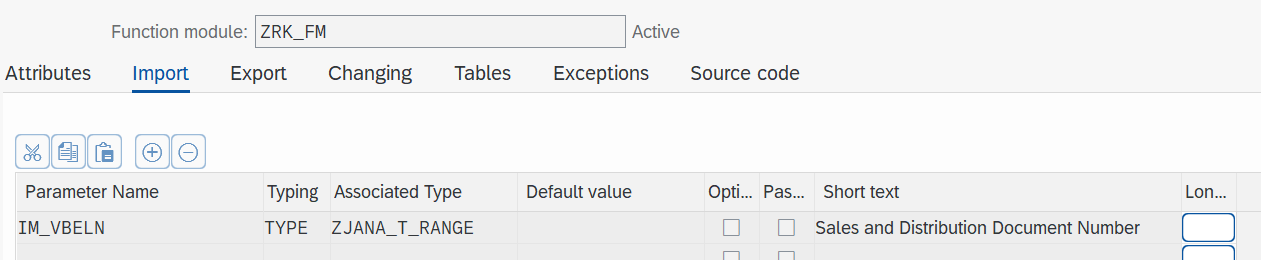


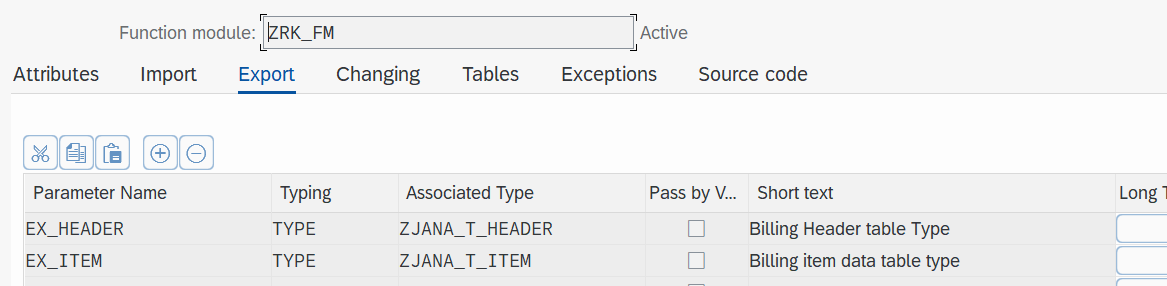
Called subroutine in below code:  


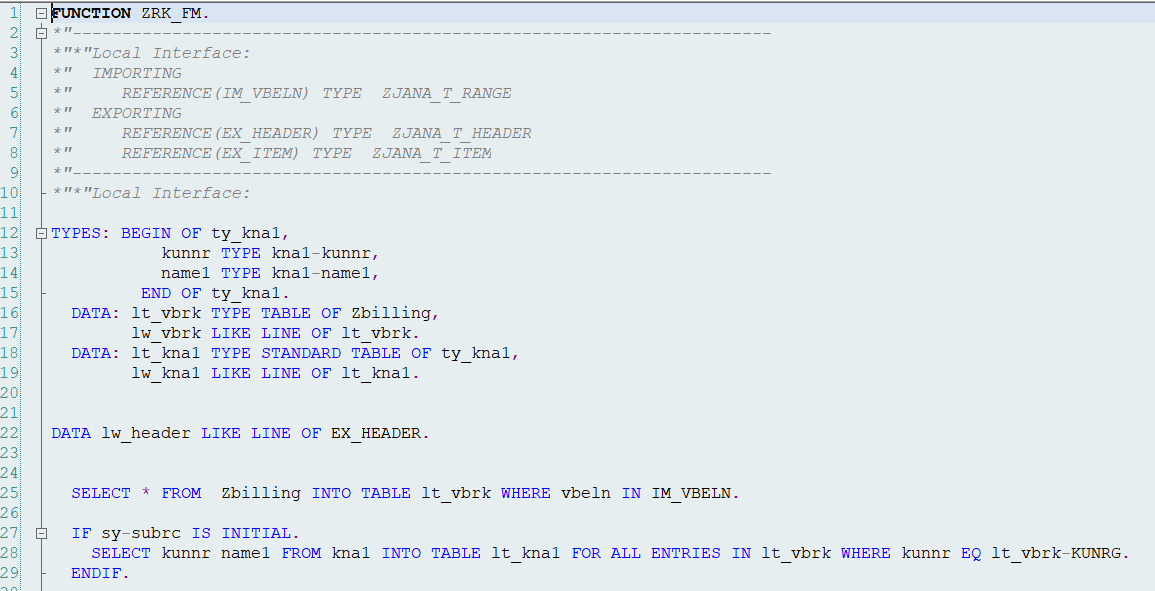
  
  
Output:  


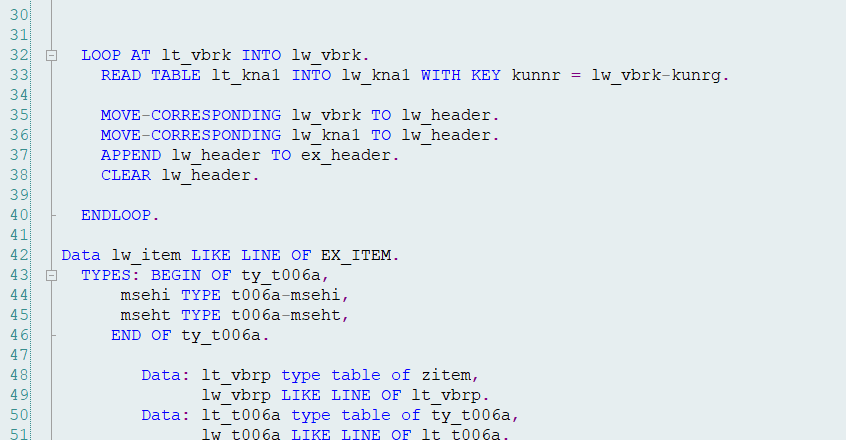


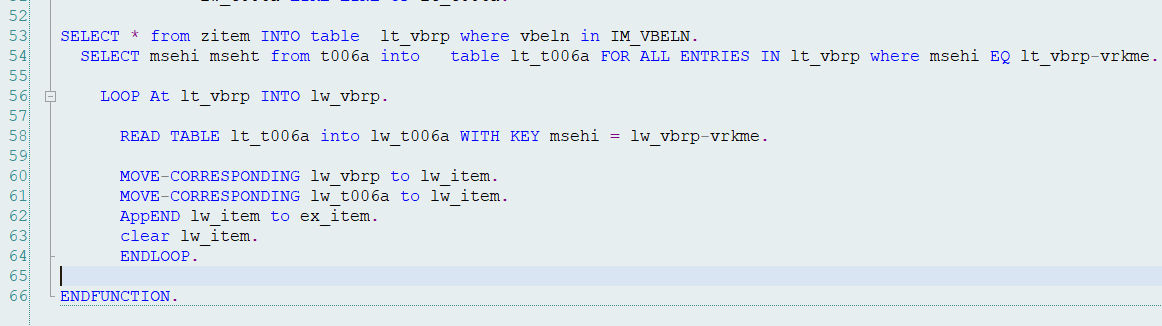
………………………………………………………………………………………………

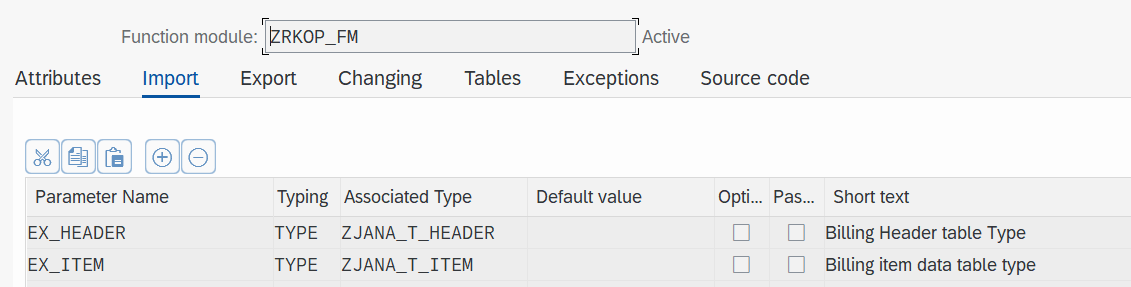
Assignment 16: Function module.  
below are function modules created and called in our executable program.  


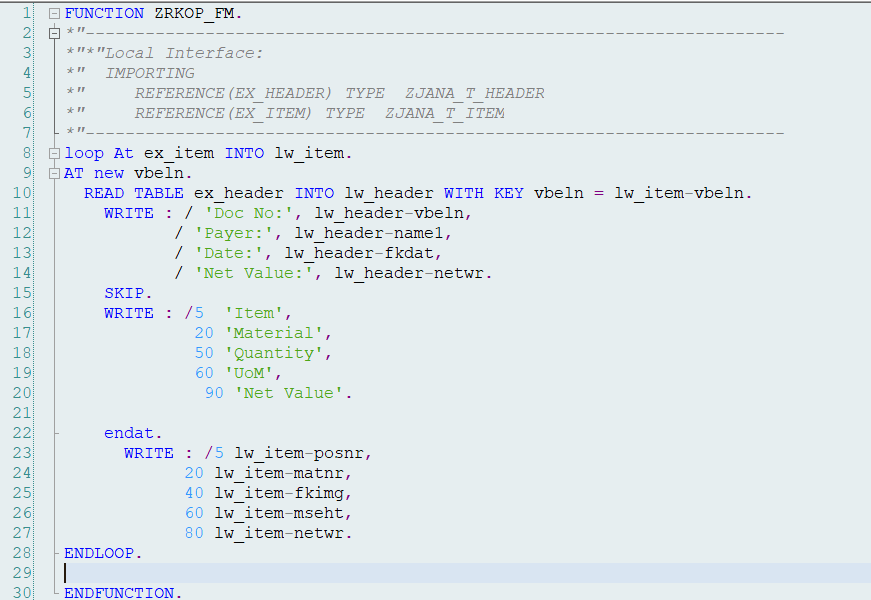


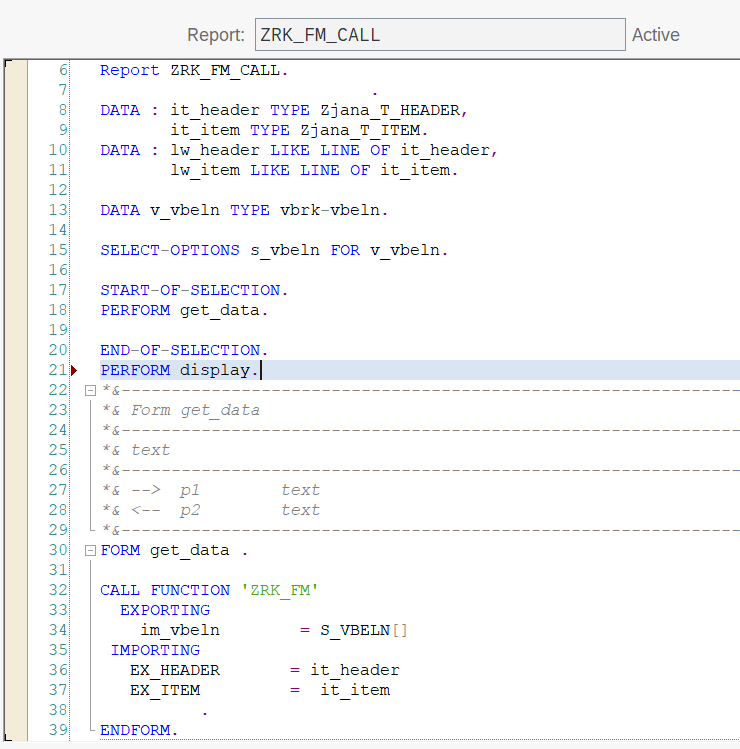


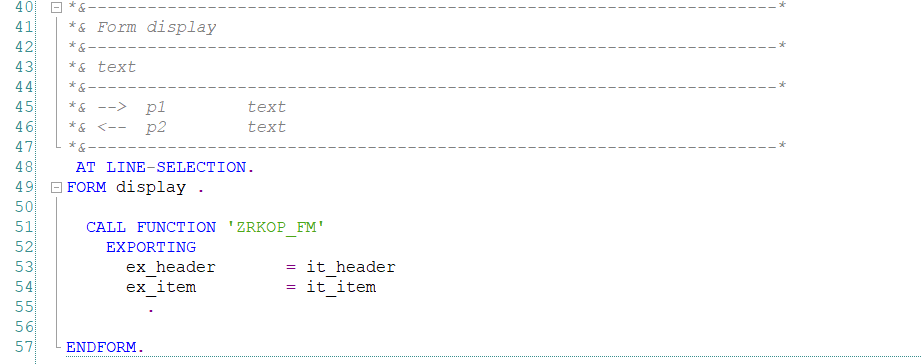




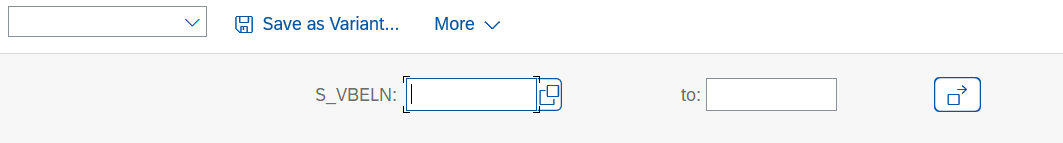


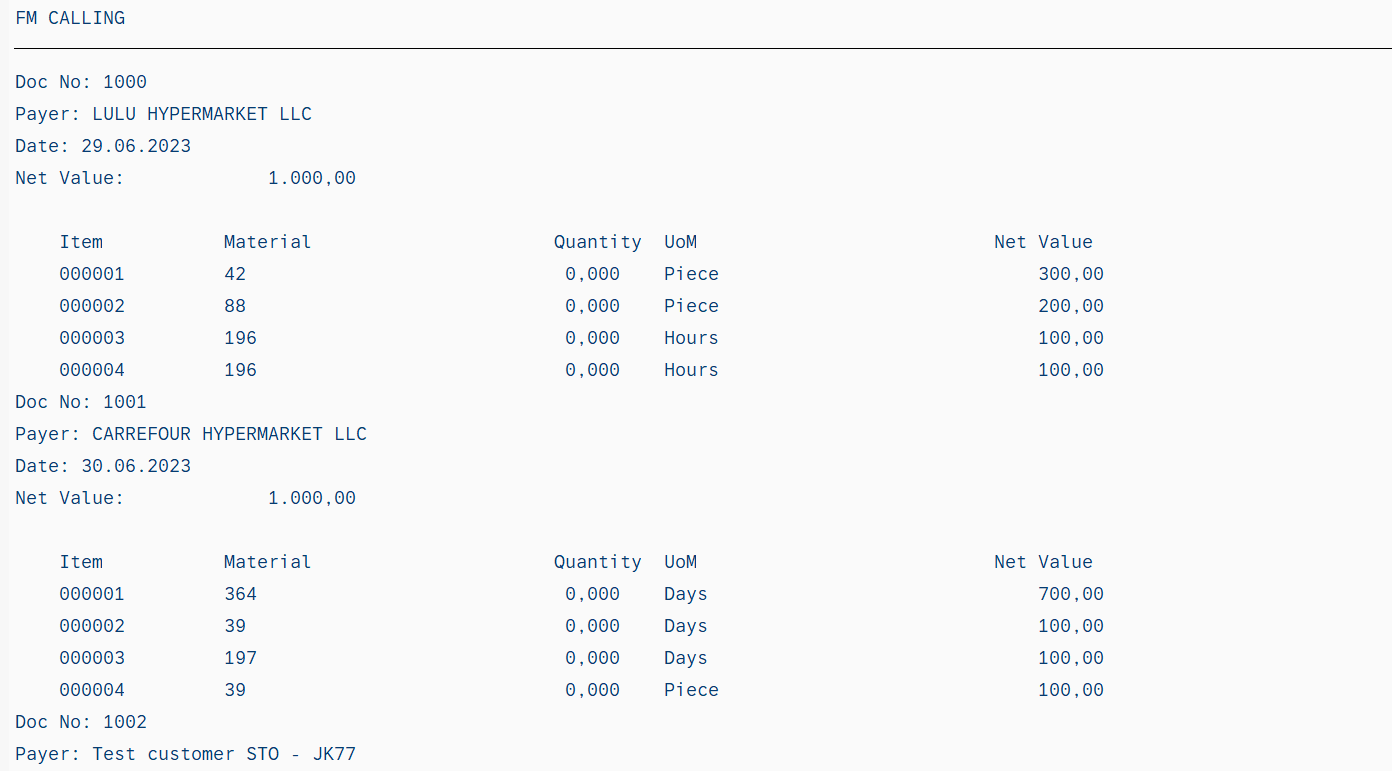


In below code we called function modules.  


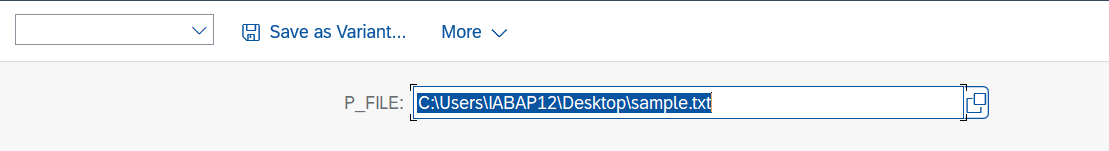


Output:





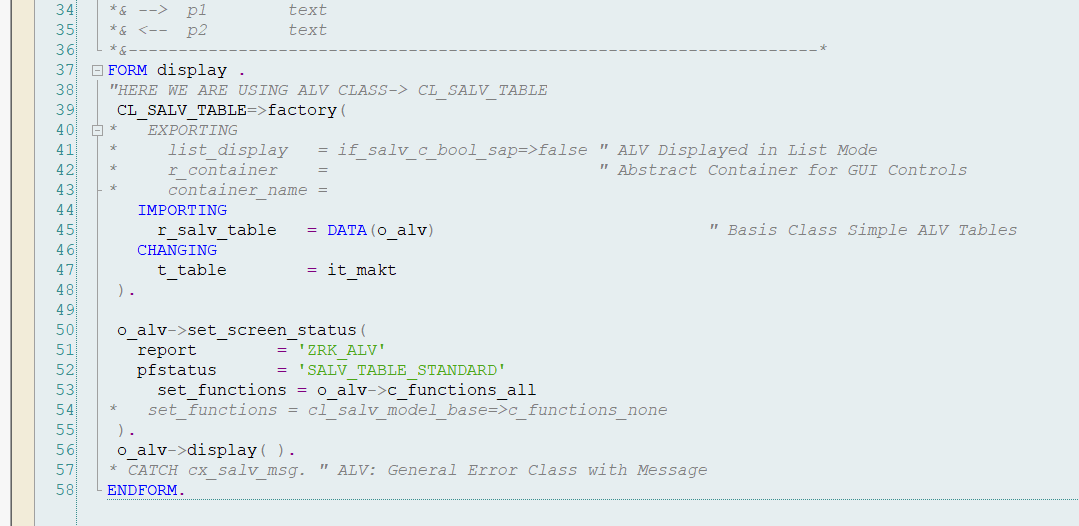
…………………………………………………………………………………………………..

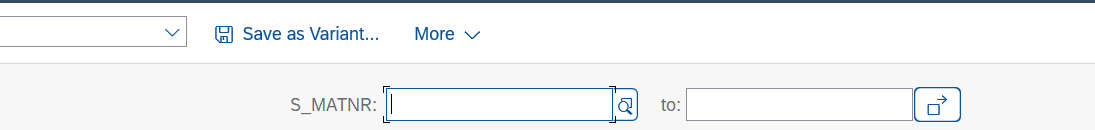
Assignment 17: File upload using: 'GUI\_UPLOAD'  
  
  
REPORT ZRK\_FILE\_UPLOAD.  
TYPES : BEGIN OF ty\_file,  
 f1 TYPE char10,  
 f2 TYPE char10,  
 f3 TYPE char10,  
 END OF ty\_file.  
  
DATA : it\_file TYPE STANDARD TABLE OF ty\_file,  
 wa\_file LIKE LINE OF it\_file.  
  
DATA v\_file TYPE string.  
PARAMETERS p\_file TYPE IBIPPARMS-PATH.*"string.*  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_file.  
 CALL FUNCTION 'F4\_FILENAME'  
 EXPORTING  
 PROGRAM\_NAME = SYST-CPROG  
 DYNPRO\_NUMBER = SYST-DYNNR  
 FIELD\_NAME = 'P\_FILE'  
 IMPORTING  
 FILE\_NAME = p\_file  
 .  
  
  
START-OF-SELECTION.  
v\_file = p\_file.  
CALL FUNCTION 'GUI\_UPLOAD'  
 EXPORTING  
 filename = v\_file  
 FILETYPE = 'ASC'  
 HAS\_FIELD\_SEPARATOR = 'X'  
*\* HEADER\_LENGTH = 0*  
*\* READ\_BY\_LINE = 'X'*  
*\* DAT\_MODE = ' '*  
*\* CODEPAGE = ' '*  
*\* IGNORE\_CERR = ABAP\_TRUE*  
*\* REPLACEMENT = '#'*  
*\* CHECK\_BOM = ' '*  
*\* VIRUS\_SCAN\_PROFILE =*  
*\* NO\_AUTH\_CHECK = ' '*  
*\* IMPORTING*  
*\* FILELENGTH =*  
*\* HEADER =*  
 tables  
 data\_tab = it\_file  
*\* CHANGING*  
*\* ISSCANPERFORMED = ' '*  
 EXCEPTIONS  
 FILE\_OPEN\_ERROR = 1  
 FILE\_READ\_ERROR = 2  
 NO\_BATCH = 3  
 GUI\_REFUSE\_FILETRANSFER = 4  
 INVALID\_TYPE = 5  
 NO\_AUTHORITY = 6  
 UNKNOWN\_ERROR = 7  
 BAD\_DATA\_FORMAT = 8  
 HEADER\_NOT\_ALLOWED = 9  
 SEPARATOR\_NOT\_ALLOWED = 10  
 HEADER\_TOO\_LONG = 11  
 UNKNOWN\_DP\_ERROR = 12  
 ACCESS\_DENIED = 13  
 DP\_OUT\_OF\_MEMORY = 14  
 DISK\_FULL = 15  
 DP\_TIMEOUT = 16  
 OTHERS = 17  
 .  
IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
ENDIF.  
  
LOOP AT it\_file INTO wa\_file.  
WRITE : / wa\_file-f1, wa\_file-f2, wa\_file-f3.  
ENDLOOP.  


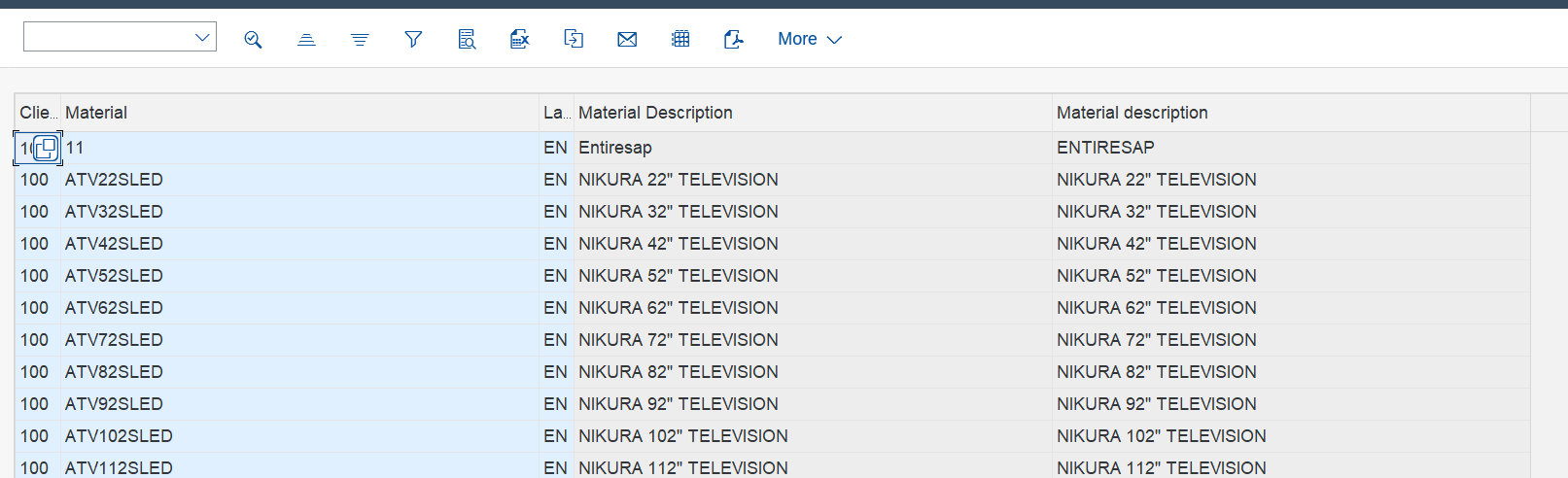


………………………………………………………………………………………………………

Assignment: CLASS ALV: Displaying material, material description, language using class ALV.  





………………………………………………………………………………………………………

Function module ALV: Displaying material number, description, language using function module ALV.  
  

