### **MAFD 6202 - Assignment 05**

For this assignment, you will be setting up an updated Inquiry program, new Invoice Summary program, and small change to Assignment 2 Menu program and Menu BMS for new Invoice Summary program. There will be small modifications to the cobol and BMS map to allow you to have individual copies of the program running at the same time, and use individual customer files for the program to use.

Invoice Summary Program:

Program Overview

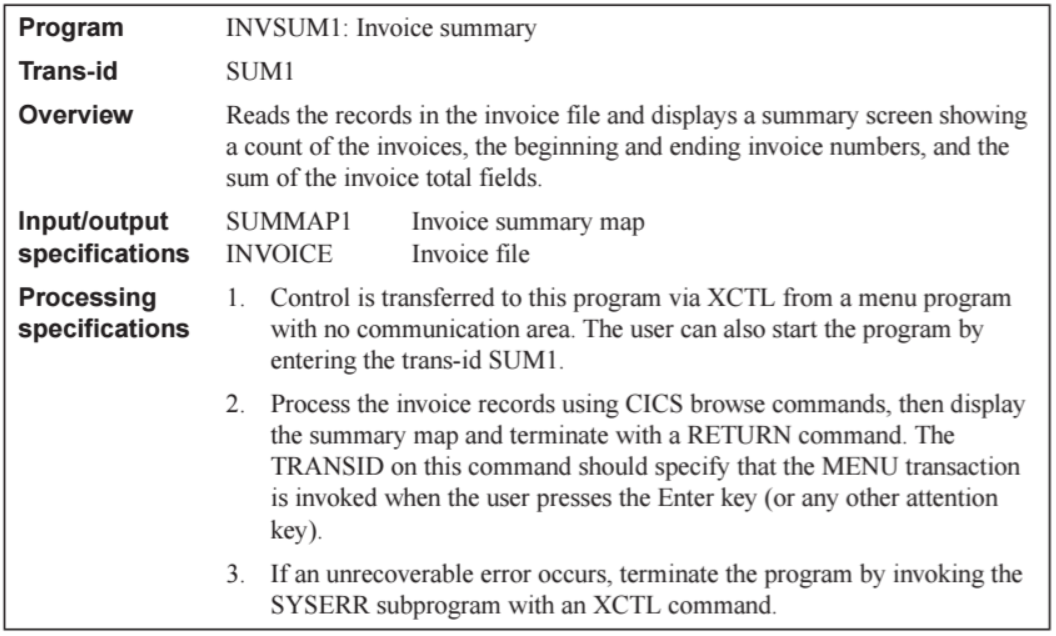
A simple, non-interactive program that browses a file of invoice records and displays information summarized from the file. The summary display shows the number of invoices in the file, the first and last invoice numbers, and the sum of the total values of all the invoices in the file.  
The invoice summary program doesn’t interact with a user, because it is conversational rather than pseudo-conversational. The structure is more like what you would expect for a batch COBOL report-preparation program than for a CICS program.

The structure chart shows that module 0000 calls module 1000 to start the browse. Then, it calls module 2000 repeatedly to read records sequentially from the invoice file. When the end of the file has been reached, module 0000 calls module 3000 to end the browse and module 4000 to send the summary display to the terminal.

The mapset for this program is unusual in that the DFHMSD macro is coded with the MODE=OUT parameter. It could have been coded MODE=INOUT, but that is not necessary because the map doesn’t accept any input. This program isn’t typical of CICS programs, since most require user input and are pseudo-conversational rather than conversational.

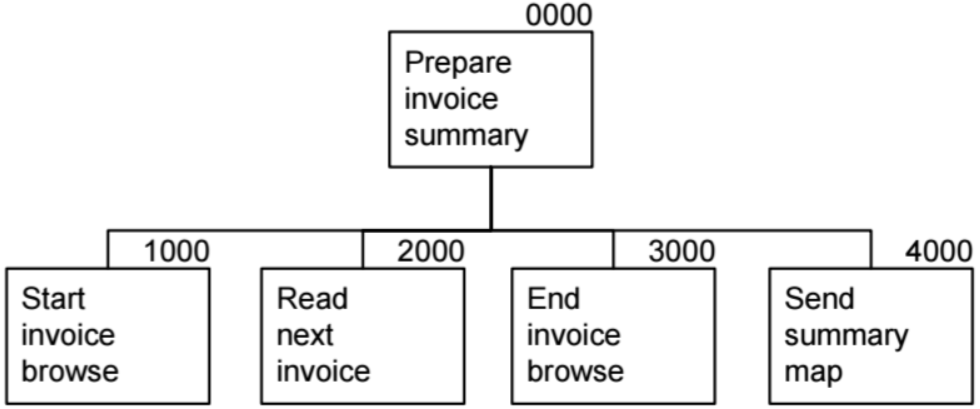
The code for the symbolic map shows the result of coding MODE=OUT, and the copy member consists of a single 01-level item that defines the output map. The three-byte FILLER field you usually see before each named field in an output map has been broken down into two fields: a two-byte FILLER field and a one-byte attribute field (suffix A) to allow the manipulation of the standard attributes for all the output fields.

Original Program Description:

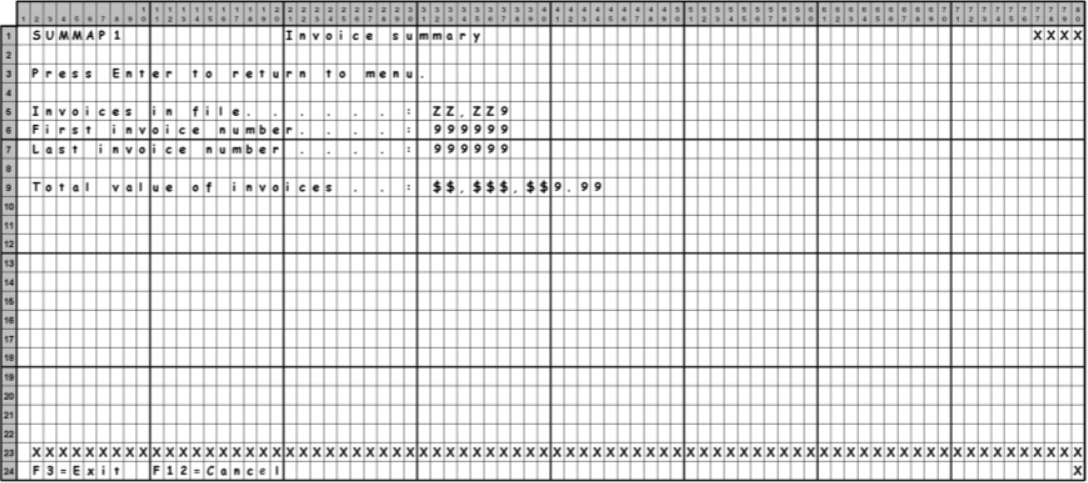


Additional Program Information:

* Structure Chart

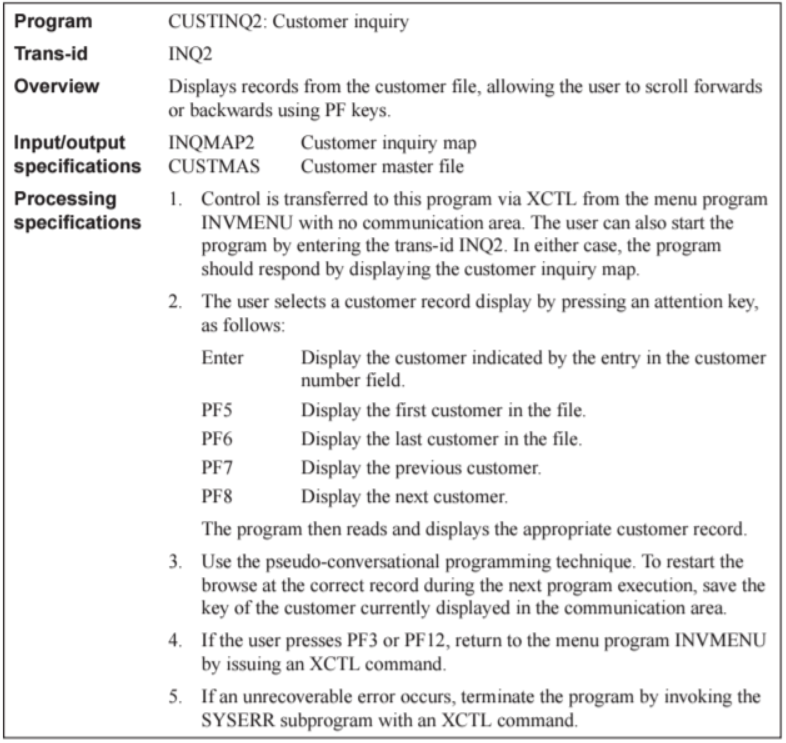


* BMS screen designs (24 lines x 80 columns)



Inquiry Program Description:

Program Overview



The customer inquiry program lets a terminal user display records from a customer file.

* Various attention keys indicate which record is displayed.
* To display a specific customer record, the user types a customer number and presses the Enter key.
* To display the first record in the file, the user presses PF5.
* To display the last record, the user presses PF6.
* To display the previous customer or the next customer, the user presses PF7 or PF8.

This is a pseudo-conversational program, so each execution that requires a browse must start a new browse operation.

* To keep track of the customer record that’s currently displayed, the program stores the customer number in the communication area between program executions.
* Then, if the user requests the next or previous customer, the program can use the customer number in the communication area to position the browse at the appropriate record.

The event/response chart for the inquiry program lists the series of commands that will be executed in response to each attention key.

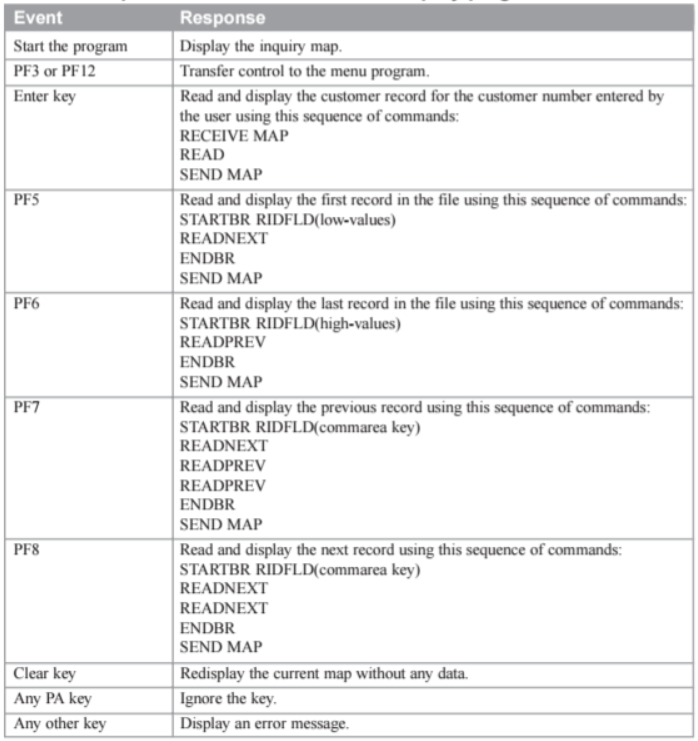
* If the user invokes the program by pressing PF5, the program responds by starting a browse at the beginning of the file, reading the first record, ending the browse, and displaying the record at the terminal.
* If the user invokes the program by pressing PF6, the program starts the browse at the end of the file, reads the last record, ends the browse, and displays the record at the terminal.

The process for displaying the previous or next customer is more complicated than for displaying the first or last customer:

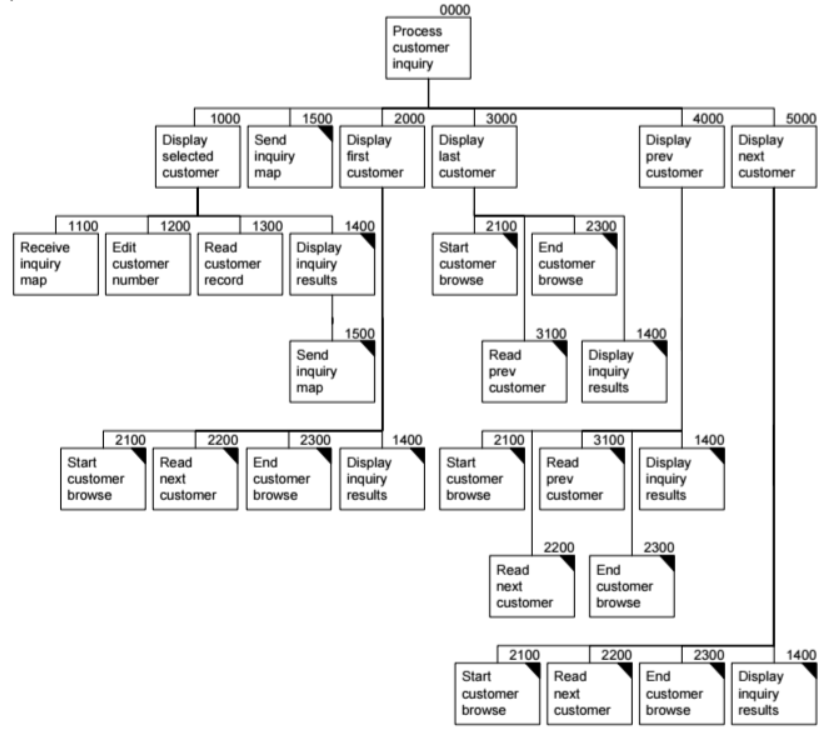
* The STARTBR command will position the browse at the customer record that’s currently displayed.
* The READNEXT command that follows will read that record.
* To display the previous customer, the program must issue two READPREV commands.
* To display the next customer, it must issue another READNEXT command.

Additional Program Information:

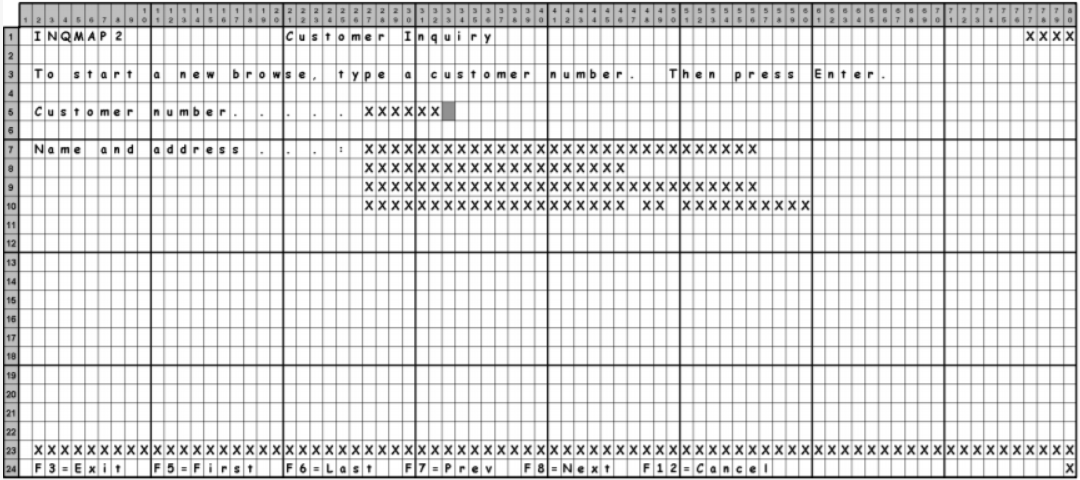
* Event Chart



* Structure Chart



* BMS screen designs (24 lines x 80 columns)



Files provided on DCConnect and/or Mainframe:

* EMPTY.BMS in KC02708.A5.BMS on mainframe
* EMPTY.CBL in KC02708.A5.COBOL on mainframe
* EMPTY.JCL in KC02708.A5.JCL on mainframe
* EMPTY.JCL in KC02708.A5.PROCLIB on mainframe
* CUSTINQ2.CBL on DCConnect  
  (EMPTY.CBL as template with this COBOL source to create CMINQYYY.CBL in KC03XXX.A5.COBOL
* INQSET2.BMS on DCConnect  
  (EMPTY.BMS as template with this BMS source to create INQSYYY.BMS in KC03XXX.A5.BMS
* INVSUM1.CBL on DCConnect  
  (EMPTY.CBL as template with this COBOL source to create INSUMYYY.CBL in KC03XXX.A5.COBOL
* SUMSET1.BMS on DCConnect  
  (EMPTY.BMS as template with this BMS source to create SUMSYYY.BMS in KC03XXX.A5.BMS

Naming Conventions:

* Your user id on the mainframe is KC03XXX
* Setup an A5 filter group in Developer with filter criteria KC03XXX.A5.\*
* While working through the instructions below   
  you will create these partitioned datasets with these members

|  |  |
| --- | --- |
| * + KC03XXX.A5.BMS |  |
| * + - INQSXXX | BMS Source for Mapset INQSXXX includes Map INQMXXX |
| * + - SUMSXXX | BMS Source for Mapset SUMSXXX includes Map SUMMXXX |
|  |  |
| * + KC03XXX.A5.COBOL |  |
| * + - CMINQXXX | COBOL source file for  Inquiry program |
| * + - INQSXXX | COBOL source map created  from BMS generate used as  COBOL COPY file |
| * + - INSUMXXX | COBOL source file for  Invoice Summary program |
| * + - SUMSXXX | COBOL source map created  from BMS generate used as  COBOL COPY file |
|  |  |
| * + - INVOICE | Invoice COBOL copy file |
|  |  |
| * + KC03XXX.A5.JCL |  |
| * + - CMINQXXX | Compile/LKED of CMINQXXX |
| * + - INQSBXXX | CICS BMS generate for INQSXXX |
| * + - INSUMXXX | Compile/LKED of INSUMXXX |
| * + - SUMSBXXX | CICS BMS generate for SUMSXXX |
|  |  |
| * + KC03XXX.A5.PROCLIB |  |
| * + - CMIFHXXX | BMS generate sub-procedure for Inquiry program |
| * + - CMIFIXXX | BMS generate sub-procedure for Invoice Summary program |

|  |
| --- |
| **The original members are available in DCConnect.   You need to create these by copying the correctly named datasets and create the correctly named members in the datasets on the mainframe to get this program working.  There are datasets with EMPTY members as templates are available under user id KC02708.**  **To create the some of the members you may need to refer to members from earlier assignments to get the correct content and syntax.** |

**Transid and File Notes:**

* Transaction ID is IXXX and SXXX  
  (I for Inquiry, S for Invoice Summary, and XXX from your user id)

Instructions:  
**Note: Refer to Assignment 1 instructions for more details about the modifications required to the data sets and members in the following steps.**

1. Copy starter data sets and empty members   
   from KC02708 to KC03XXX  
   1. KC02708.A5.BMS
   2. KC02708.A5.COBOL
   3. KC02708.A5.PROCLIB
   4. KC02708.A5.JCL
2. Generate BMS for Invoice Summary program  
   1. Use EMPTY.BMS to create BMS source SUMSXXX.BMS with BMS file SUMSET1.BMS
   2. Use EMPTY.JCL to create SUMSBXXX.JCL to generate BMS based on JCL used to generate BMS in A1 or A2 or A3 or A4
   3. Submit SUMSBXXX.JCL
   4. This will generate BMS mapset and store it in   
      TSOECCC.CICSTS12.STUDENT.LOADLIB for use with CICS
   5. This will generate COBOL COPY library SUMSXXX that matches the BMS mapset,   
      and is included in COBOL program INSUMXXX   
      using COPY SUMSXXX statement when program is compiled
   6. Ensure return code = 0 or 4  
      NOTE: RETURN CODE=0004 May be due to these warnings:  
        
      - IGYOS4090-I The "LIB" option specification is no longer required. COBOL library processing is always in effect.  
        
      - IGYOS4020-W The "NSYMBOL(NATIONAL)" option was discarded due to option conflict resolution. The "NODBCS" option from compiler invocation parameter took precedence.
3. Compile new Invoice Summary program  
   1. Use EMPTY.CBL to create INSUMXXX program member with provided COBOL program source file
   2. Use EMPTY.CBL to create correctly named copy members with provided COBOL copy files
   3. Use EMPTY.JCL to create INSUMXXX.JCL   
      to compile CICS program
   4. Ensure any copy files referenced in the COBOL are available for the compile process
   5. Ensure INVOICE file is open before any READ/WRITE/REWRITE/DELETE operation is done  
      (refer to earlier assignment instructions for source lines needed to check VSAM file status and Open the file if needed)
   6. Ensure INVOICE file is closed when PF3 or PF12 operation is requested  
      (refer to earlier assignment instructions for source lines needed to check VSAM file status and Close the file if needed)
   7. Submit INSUMXXX.JCL
   8. This will create executable for INSUMXXX program and store it in TSOECCC.CICSTS12.STUDENT.LOADLIB for use with CICS
   9. Ensure return code = 0 or 4  
      NOTE: RETURN CODE=0004 May be due to these warnings:  
        
      - IGYOS4090-I The "LIB" option specification is no longer required. COBOL library processing is always in effect.  
        
      - IGYOS4020-W The "NSYMBOL(NATIONAL)" option was discarded due to option conflict resolution. The "NODBCS" option from compiler invocation parameter took precedence.
4. Generate BMS for new Inquiry program  
   1. Use EMPTY.BMS to create BMS source INQSXXX.BMS with BMS file INQSET2.BMS
   2. Use EMPTY.JCL to create INQSBXXX.JCL to generate BMS based on JCL used to generate BMS in A1 or A2 or A3 or A4
   3. Submit INQSBXXX.JCL
   4. This will generate BMS mapset and store it in   
      TSOECCC.CICSTS12.STUDENT.LOADLIB for use with CICS
   5. This will generate COBOL COPY library INQSXXX that matches the BMS mapset,   
      and is included in COBOL program CMINQXXX   
      using COPY INQSXXX statement when program is compiled
5. Ensure return code = 0 or 4  
   NOTE: RETURN CODE=0004 May be due to these warnings:  
     
   - IGYOS4090-I The "LIB" option specification is no longer required. COBOL library processing is always in effect.  
     
   - IGYOS4020-W The "NSYMBOL(NATIONAL)" option was discarded due to option conflict resolution. The "NODBCS" option from compiler invocation parameter took precedence.
6. Compile new Inquiry program  
   1. Use EMPTY.CBL to create CMINQXXX program member with provided COBOL program source file
   2. Use EMPTY.CBL to create correctly named copy members with provided COBOL copy files
   3. Use EMPTY.JCL to create CMINQXXX.JCL   
      to compile CICS program
   4. Ensure any copy files referenced in the COBOL are available for the compile process
   5. Ensure CUSTMAS file is open before any READ/WRITE/REWRITE/DELETE operation is done  
      (refer to earlier assignment instructions for source lines needed to check VSAM file status and Open the file if needed)
   6. Ensure CUSTMAS file is closed when PF3 or PF12 operation is requested  
      (refer to earlier assignment instructions for source lines needed to check VSAM file status and Close the file if needed)
   7. Submit CMINQXXX.JCL
   8. This will create executable for CMINQXXX program and store it in TSOECCC.CICSTS12.STUDENT.LOADLIB for use with CICS
7. Ensure return code = 0 or 4  
   NOTE: RETURN CODE=0004 May be due to these warnings:  
     
   - IGYOS4090-I The "LIB" option specification is no longer required. COBOL library processing is always in effect.  
     
   - IGYOS4020-W The "NSYMBOL(NATIONAL)" option was discarded due to option conflict resolution. The "NODBCS" option from compiler invocation parameter took precedence.
8. Update Menu BMS and MENU Program from A2 to add option 4 to transfer control to Invoice Summary program  
   1. Update MENU BMS to include text on screen   
      for selection 4 that transfers control to INSUMXXX program  
      (This BMS should remain in KC03XXX.A2 datasets   
      versus moving it to KC03XXX.A5 datasets for this change)
   2. Recompile MENU BMS   
      1. Ensure return code = 0 or 4
   3. Update MENU Program to handle INSUMXXX for selection 4  
      (This Program should remain in KC03XXX.A2 datasets   
      versus moving it to KC03XXX.A5 datasets for this change)
   4. Recompile MENU Program  
      1. Ensure return code = 0 or 4
   5. INSTALL BMS and Program in CICS   
      using CEDA DISPLAY command  
      (not required to re-define objects in CICS)
9. Setup objects in CICS for new Inquiry and Invoice Summary programs
10. Define Mapsets
11. Define Programs
12. Define Transaction
13. Check objects you have defined
14. Install MAPSET, PROGRAM, TRANSACTION objects
15. Test new Inquiry and Invoice Summary program
16. Choose from Menu the Inquiry program
17. Test Inquiry functionality including the new Browse functions
18. Choose from Menu the Invoice Summary program  
    to see the summary information displayed
19. Return to Menu when completed
20. Create another order
21. Repeat c and d to see changes in summary information displayed

**Note:   
  
If the Invoice Summary program fails with ASRA ABEND, then there might be corruption in the INVYYY file from your earlier testing of the program.   
   
- Delete the INVYYY vsam file(s)  
 (.INVYYY main file, .INVYYY.data, and .INVYYY.index)   
  
- Delete the INVYYY flat file if FL JCL is not already doing this  
  
- Rerun the INVCLYYY JCL to recreate an empty vsam file  
  
- Rerun the INVLOYYY JCL to load the initial data into the vsam file  
  
- Rerun the INVFLYYY JCL to see the initial data in the FLAT file  
  
- Test the Invoice Summary program by using the SYYY transaction id to see the summary information displayed  
(make a note of the data displayed)  
  
- If it does not work then your problem may be something else   
and you should continue troubleshooting the problem  
  
- If it works then create a new order with ORDERYYY  
(make a note of the new order details)  
  
- Retest Invoice Summary program by using the SYYY  
to see the changes in the summary information displayed   
that should include the new order details**

**Submission and Marking:**

**Inquiry BMS**

BMS Source file that includes mapset and map name changes

and changes made to rename mapset and map for your userid – 4 marks

BMS JCL file that includes changes for your userid – 2 marks

BMS copy file generated by BMS JCL that is used in Maint program COBOL compile – 2 marks

BMS entire job summary including compile listing with return code 0 or 4 – 2 marks

**Inquiry BMS – 10 marks**

**Inquiry Program**

COBOL Source file that includes provided program improvements for VSAM Inquire/Open/Close   
and for PF3 and PF12, browse functionality using PF5/PF6/PF7/PF8, and any changes for your userid to TransID, Program, Mapset, Map, File objects used with CICS – 20 marks

COBOL Compile/LKED JCL that includes changes for your userid – 2.5 marks

COBOL Translate JCL from PROCLIB dataset that includes changes for your userid – 2.5 marks

COBOL entire job summary including the compile listing with return code 0 or 4 – 5 marks

**Inquiry Program – 30 marks**

**Invoice Summary BMS**

BMS Source file that includes mapset and map name changes

and changes made to rename mapset and map for your userid – 4 marks

BMS JCL file that includes changes for your userid – 2 marks

BMS copy file generated by BMS JCL that is used in Maint program COBOL compile – 2 marks

BMS entire job summary including compile listing with return code 0 or 4 – 2 marks

**Invoice Summary BMS – 10 marks**

**Invoice Summary Program**

COBOL Source file that includes provided program improvements for   
VSAM Inquire/Open/Close, and any changes for your userid to   
TransID, Program, Mapset, Map, File objects used with CICS – 20 marks

COBOL Compile/LKED JCL that includes changes for your userid – 2.5 marks

COBOL Translate JCL from PROCLIB dataset that includes changes for your userid – 2.5 marks

COBOL entire job summary including the compile listing with return code 0 or 4 – 5 marks

**Invoice Summary Program – 30 marks**

**Screenshots of CEDA DISPLAY – GROUP DUR5202**– use \*XXX in each file type to get screen showing \*INSTALL of

1. four Files – 5 marks for each entry on screen  
   (INSTALL not required for FILE objects, since the INSTALL was done earlier)
2. five Mapsets – 5 marks for each entry on screen
3. six Programs – 5 marks for each entry on screen
4. six Transactions – 5 marks each entry on screen

**Setup on CICS – 105 marks**

**Screenshots of a successful manual open and close of each VSAM file   
using CEMT SET FILE(FFFXXX) OPEN and CEMT SET FILE(FFFXXX) CLOSE CICS commands  
(5 marks each for open screenshot, and 5 marks each for close screenshot)**

1. Customer Master file (two screenshots – one for each of open and close)  
   (also called CUSTMAS, and known in your implementation as CMFXXX)
2. Invoice file (two screenshots – one for each of open and close)  
   (also called INVOICE, and known in your implementation as INVXXX)
3. Product file (two screenshots – one for each of open and close)  
   (also called PRODUCT, and known in your implementation as PRDXXX)
4. Invoice Control file (two screenshots – one for each of open and close)  
   (also called INVCTL, and known in your implementation as CTLXXX)

**VSAM Open/Close screenshots – 40 marks**

**Screenshots of new Inquiry and Invoice Summary program showing these cases   
(5 marks each)**

1. Inquiry screen with error message after entering blanks in Customer Number field
2. Inquiry screen with ‘Customer does not exist’ message for customer number not in the file
3. Inquiry screen with valid data after entering ‘123456’ or any other valid customer number
4. Inquiry screen after function 5 (PF5) First customer
5. Inquiry screen after function 8 (PF8) Next customer
6. Inquiry screen after function 6 (PF6) Last customer
7. Inquiry screen after function 7 (PF7) Previous customer
8. Invoice Summary screen   
   (this will be either the summary for the data from the initial load,   
   or include some extra orders you have created)
9. Invoice Summary screen after adding another order(s)  
   (this will be summary for step 9 plus any orders you have done since step 8)

**Inquiry and Invoice Summary Programs working on CICS – 45 marks**

**Testing of updated Menu, updated Inquiry, Maint, Order Entry,   
Get Invoice Number, and Invoice Summary programs during marking   
– 120 marks (20 marks per program)**

**Total 390 marks**