Session - 3

Objectives

- Introduction to BMS
- Physical and Symbolic Map
- Map and Mapset
- Map Definition Macros
- Screen Design Considerations

Introduction To BMS
(Basic Mapping Support)

- Primary functions of BMS
  - Removal of device dependent codes from Application Program
  - Removal of constant information from Application program (Headers, Titles...)
  - Construct NMDS - Native Mode Data Stream
  - Text handling
  - Terminal Paging & Message routing
  - Contents of the screen defined thru' BMS is called Map.
  - Map is a program written in assembly language.
  - BMS macros are available for Map coding.

Map & Mapset

- Representation of one screen format is called Map (screen panel).
- One or more maps, linkedited together, makes up a Mapset (load module).

Types of Maps

There are 2 types of MAPS

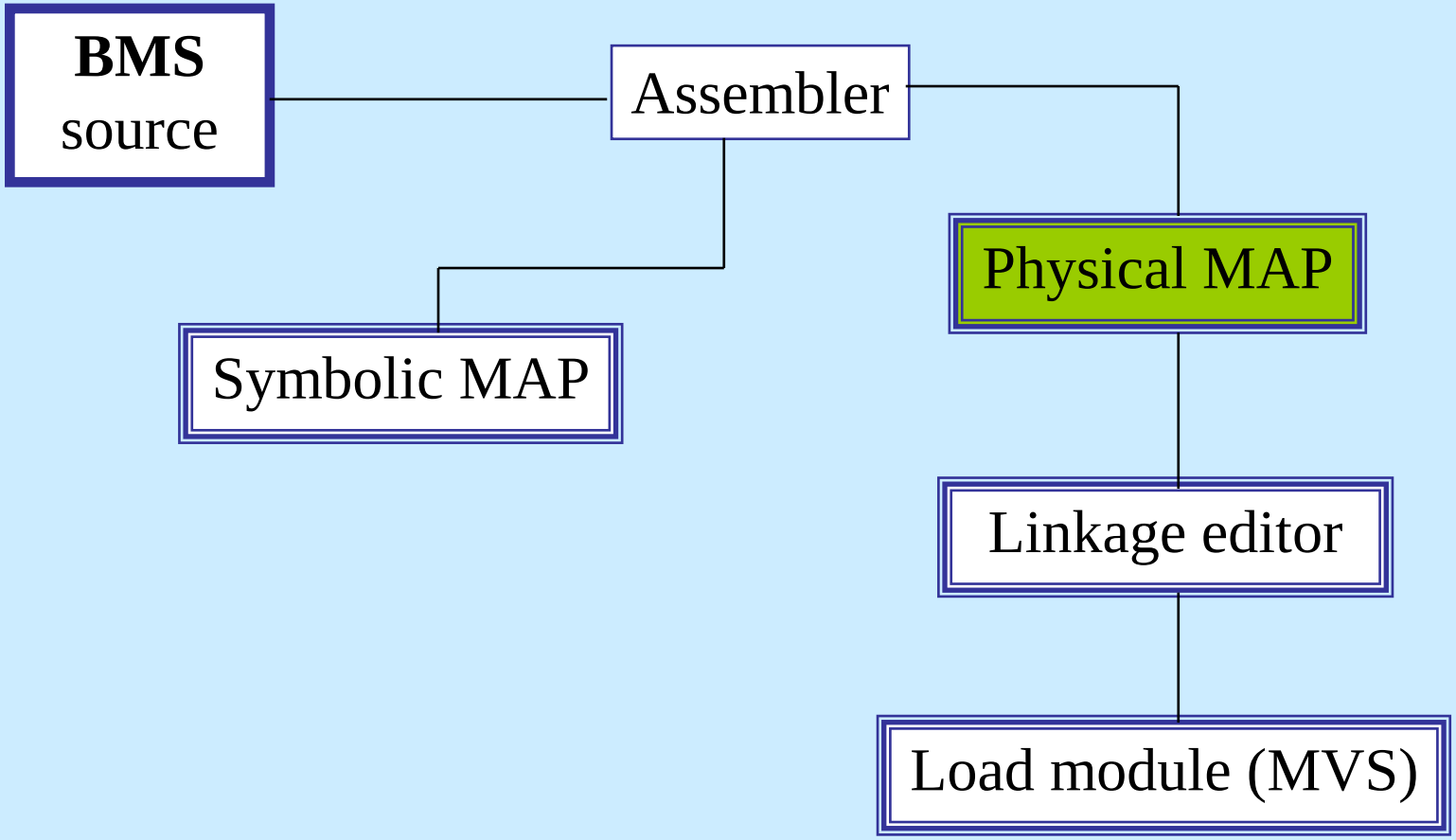
- **Physical Map**
- **Symbolic Map**

Example of Symbolic Map

```
01 EMPRECI.
    02 FILLER        PIC   X(12).
    02 EMPNAL        PIC   S9(4) COMP.
    02 EMPNAF        PIC   X.
    02 FILLER REDEFINES EMPNAF.
       03 EMPNAA     PIC X.
    02 EMPNAI        PIC  X(21).

 01 EMPRECO REDEFINES EMPRECI.
    02 FILLER        PIC X(12).
    02 FILLER        PIC X(03).
    02 EMPNAO        PIC X(21).
```

Physical & symbolic Map – logic flow



Map Definition Macros

General Format

1	10	16	72
setname	operation	operands	contd.

Example

```
EMPMAP DFHMSD    TYPE=MAP,                X
                 MODE=INOUT,              X
                 LANG=COBOL,              X
                 STORAGE=AUTO,            X
                 TIOAPFX=YES
*
*    ANY COMMENTS
```

Order of Macros

- DFHMSD TYPE=DSECT Mapset
- DFHMDI Map
- DFHMDF A field
- DFHMDF A field
- :
- DFHMDI Map
- DFHMDF A field
- DFHMDF A field
- :
- DFHMSD TYPE=FINAL Mapset
- END

DFHMSD Macro

- Define a mapset and its characteristics or to end a mapset definition
- Only one mapset is allowed in one assembly run.
- Example

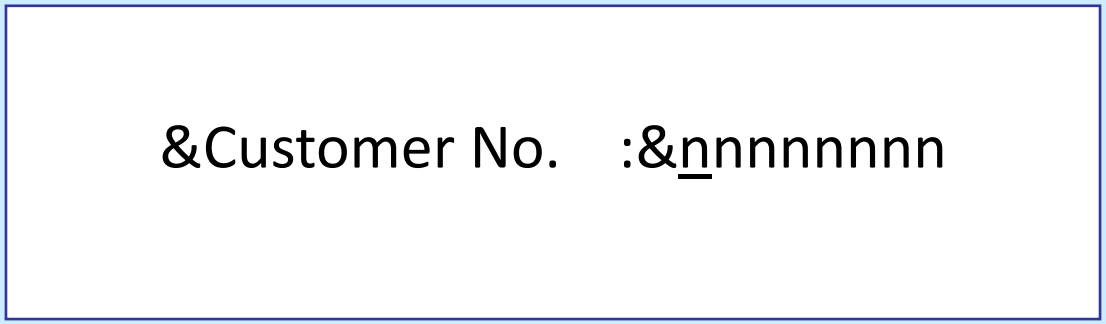
```
EMPMSET DFHMSD    TYPE=&SYSPARM ,                X
                     MODE=INOUT,                 X
                     LANG=COBOL,                 X
                     STORAGE=AUTO,               X
                     TIOAPFX=YES,                X
                     CTRL=(FREEKB,FRSET,PRINT)
```

DFHMDI Macro

- Define a map and its characteristics
- Example

```
EMPMAP DFHMDI SIZE=(ll,cc),                    X
             LINE=nn,                          X
            COLUMN=mm,                         X
            JUSTIFY=LEFT/RIGHT
```

Screen Layout



- Where
  - & Attribute character
  - n Unprotected numeric
  - - Cursor

DFHMDF macro for the above layout

- Define a field and its characteristics

- Example

```
        DFHMDF  POS(ll,cc),                                X
                INITIAL='Customer No. :',                  X
                ATTRB=ASKIP,                               X
                LENGTH=14

CUSTNO  DFHMDF POS=(ll,cc),                                X
                ATTRB=(UNPROT,NUM,FSET,IC),                X
                JUSTIFY=RIGHT,                       X
                PICIN='9(8)',                              X
                PICOUT='9(8)',                             X
                LENGTH=8
```

Attribute character

- Invisible one byte character

- Defines the characteristics of a field

  Thru' ATTRB param. of DFHMDF.

Modified Data Tag

- Indicates the field has been modified or not
- Effective use of MDT reduces the amount of data traffic.
- MDT setting/resetting
  - when the user modifies a field on the screen
  - CTRL=FRSET, defined in map/mapset
  - FSET in ATTRB parameter of DFHMDF

Dynamic Attribute Assignment

- 0 1 2 3 4 5 6 7
- 0 and 1 – Dependent bits(0 on 2 & 1 on 7)
- 2 and 3 – Attribute
  - 2 – 0 protected 1 unprotected
  - 3 – 0 alphanumeric 1 numeric
- 4 and 5 – Intensity
  - 0 0 (norm/non detectable)
  - 0 1(norm/detectable)
  - 1 0(BRT)
  - 1 1(DRK)

Skipper Technique

- Skipper technique is used to skip the cursor to the next unprotected field after one unprotected field.
- Unlabelled 1-byte field with the **autoskip** attribute
- **DFHMDF POS(ll,cc),ATTRB=ASKIP,LENGTH=1**

Stopper Technique

- To stop the cursor in order to prevent erroneous field overflow by terminal user the stopper technique can be used.
- Unlabelled 1-byte field with the **protect** attribute
- **DFHMDF POS(ll,cc),ATTRB=PROT,LENGTH=1**

Cursor Positioning Techniques

- Static positioning (map definition)
- Dynamic/Symbolic Positioning (app.pgm)
- Dynamic/Relative Positioning (app. pgm)
- Checking Cursor Position by EIBCPOSN.

Static positioning (map definition)

IC in ATTRB parameter of DFHMDF

If ‘IC’ is used with more than one field, the cursor will appear in the last field.

Dynamic/Symbolic Positioning (application program)

Move -1 to the field-length field and

SEND map with CURSOR option

Dynamic/Relative Positioning (application program)

SEND with CURSOR(data-value) option.

Numeric Sign / Decimal Point Handling

- **Numeric Sign :** For input operations, Separate fields or CR/DR field approach can be used and for output operations, PICOUT parameter can be given in macro

- **Decimal Point :** For input operations, Virtual decimal point or Separate fields approach can be used and for output operations, PICOUT parameter has to be given in the field definition macro.

Send Map

- EXEC CICS SEND
- MAP(‘map name’)
- [MAPSET(‘mapset name’)]
- [FROM(data-area)]
- [LENGTH(data_value)]
- [DATAONLY]
- [MAPONLY]
- [ERASE/ERASEAUP]
- [FREEKB]
- [FRSET]
- END-EXEC

Conditions : INVREQ,LENGERR

RECEIVE Map

- To receive input from a terminal

- Syntax :

  EXEC CICS **RECEIVE MAP** (mapname)
  [INTO(dataname)]
  [LENGTH(msg-len)]
  [MAPSET(mapsetname)]
  [HANDLE | NOHANDLE
  [RESP()]]
  END-EXEC

Conditions: INVREQ, MAPFAIL

FORMAT OF SYMBOLIC MAP

- **A 12-byte TIOA (Terminal Input/Output Area) prefix.**
- **The mapnames are suffixed with ‘I’ and ‘O’**
- **When performing INPUT functions fields suffixed with ‘L’, ’F’ and ‘I’ are meaningful**
- **When performing OUTPUT functions fields suffixed with ‘A’, and ‘O’ are meaningful**

Recap

- What is a BMS?
- What are the two kinds of maps? Why do we need them?
- What are the macros used to define?
- What is MDT, FSET and FRSET?
- What are the symbolic map fields generated?
- What are the cursor positioning techniques??

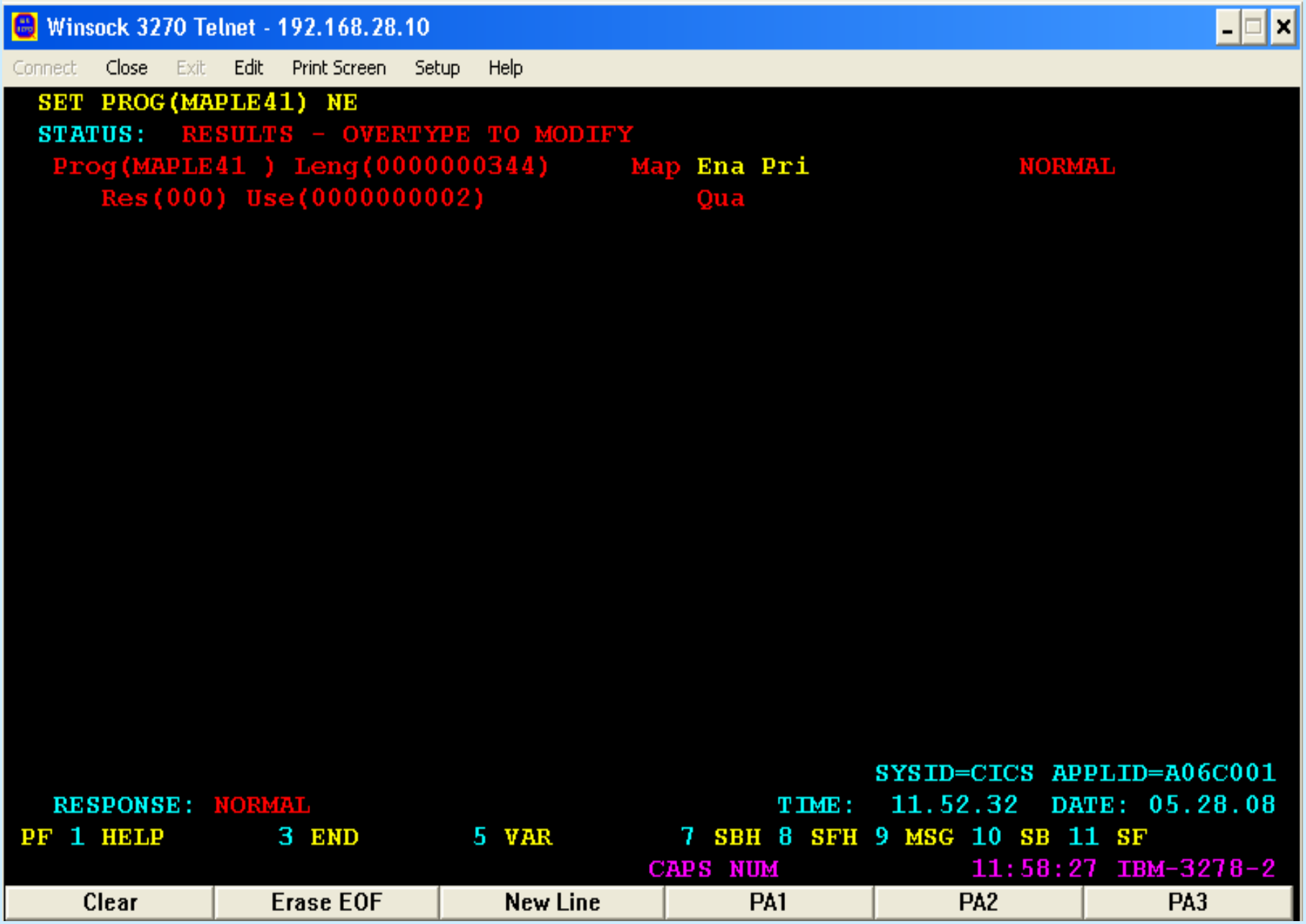
Lab Session

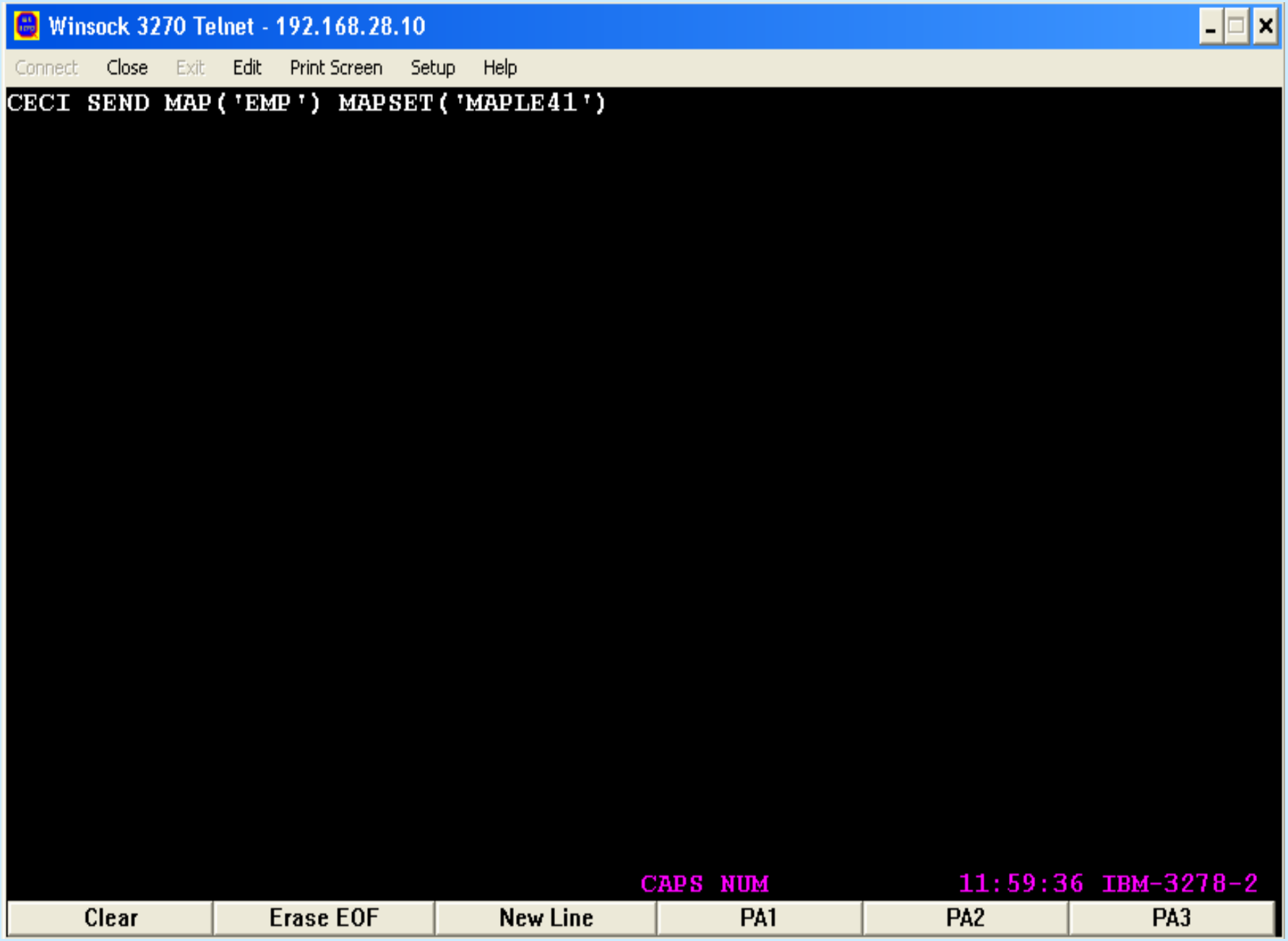
Macro Coding For map design

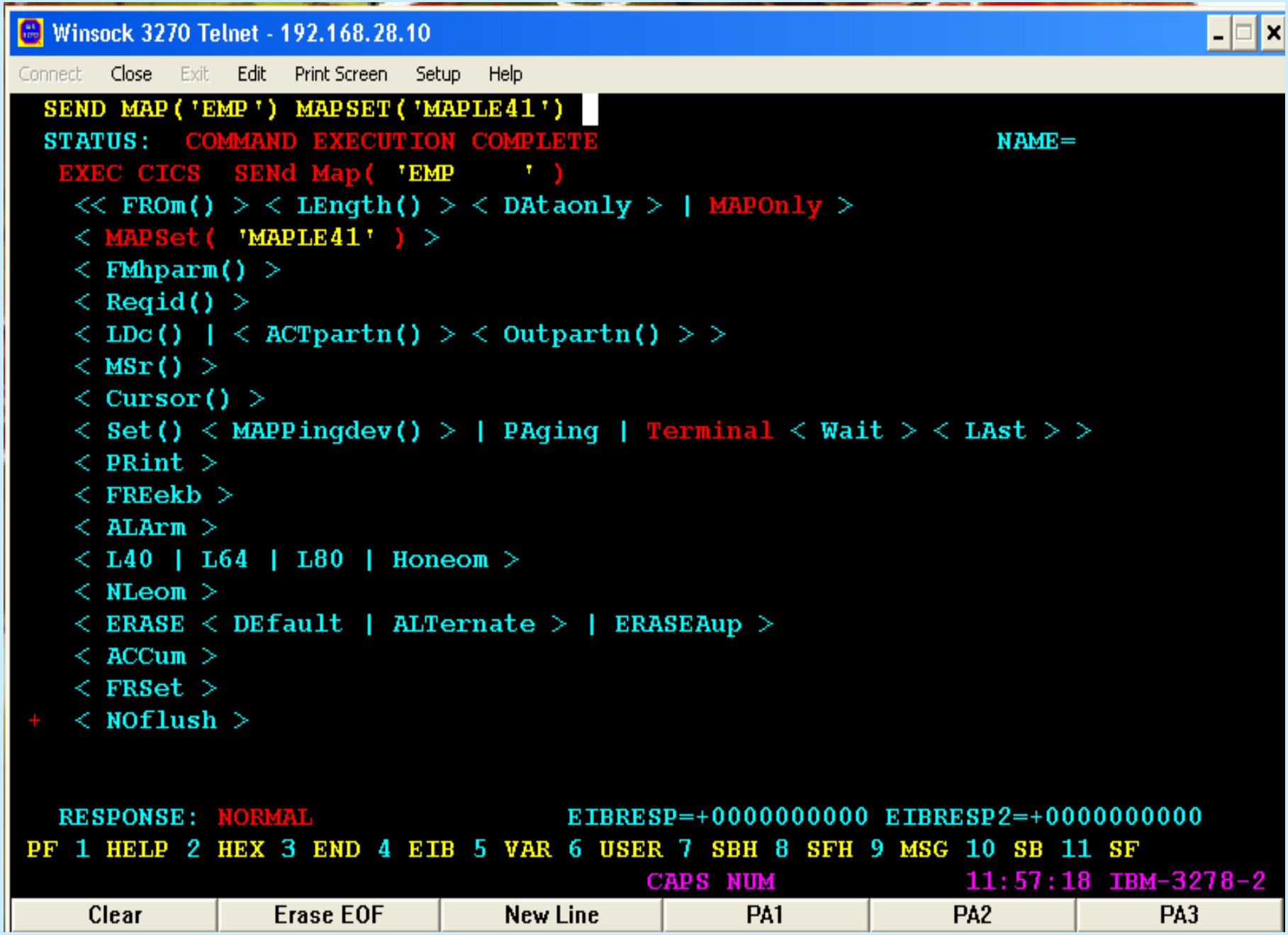
```
000001 MAPLE41  DFHMSD TYPE=&SYSPARM,LANG=COBOL,STORAGE=AUTO,
X
000002                                  TIOAPFX=YES,MODE=INOUT
000003 EMP         DFHMDI  SIZE=(24,80),LINE=1,COLUMN=1,CTRL=(FREEKB,FRSET)
000004              DFHMDF POS=(5,20),ATTRB=(NORM,ASKIP),LENGTH=16,
X
000005                          INITIAL='EMPLOYEE DETAILS'
000006              DFHMDF POS=(9,15),ATTRB=(NORM,ASKIP),LENGTH=20,
X
000007                          INITIAL='EMPLOYEE NUMBER   :  '
000008 ENO         DFHMDF POS=(9,36),ATTRB=(NORM,UNPROT,IC,FSET),LENGTH=4,
X
000009                          INITIAL='----'
000010              DFHMDF POS=(11,15),ATTRB=(NORM,ASKIP),LENGTH=20,
X
000011                          INITIAL='EMPLOYEE  NAME       : '
000012 ENAME    DFHMDF  POS=(11,36),ATTRB=(NORM,UNPROT,FSET),LENGTH=15,
X
000013                          INITIAL='---------------'
000014              DFHMSD  TYPE=FINAL
000015                           END
```

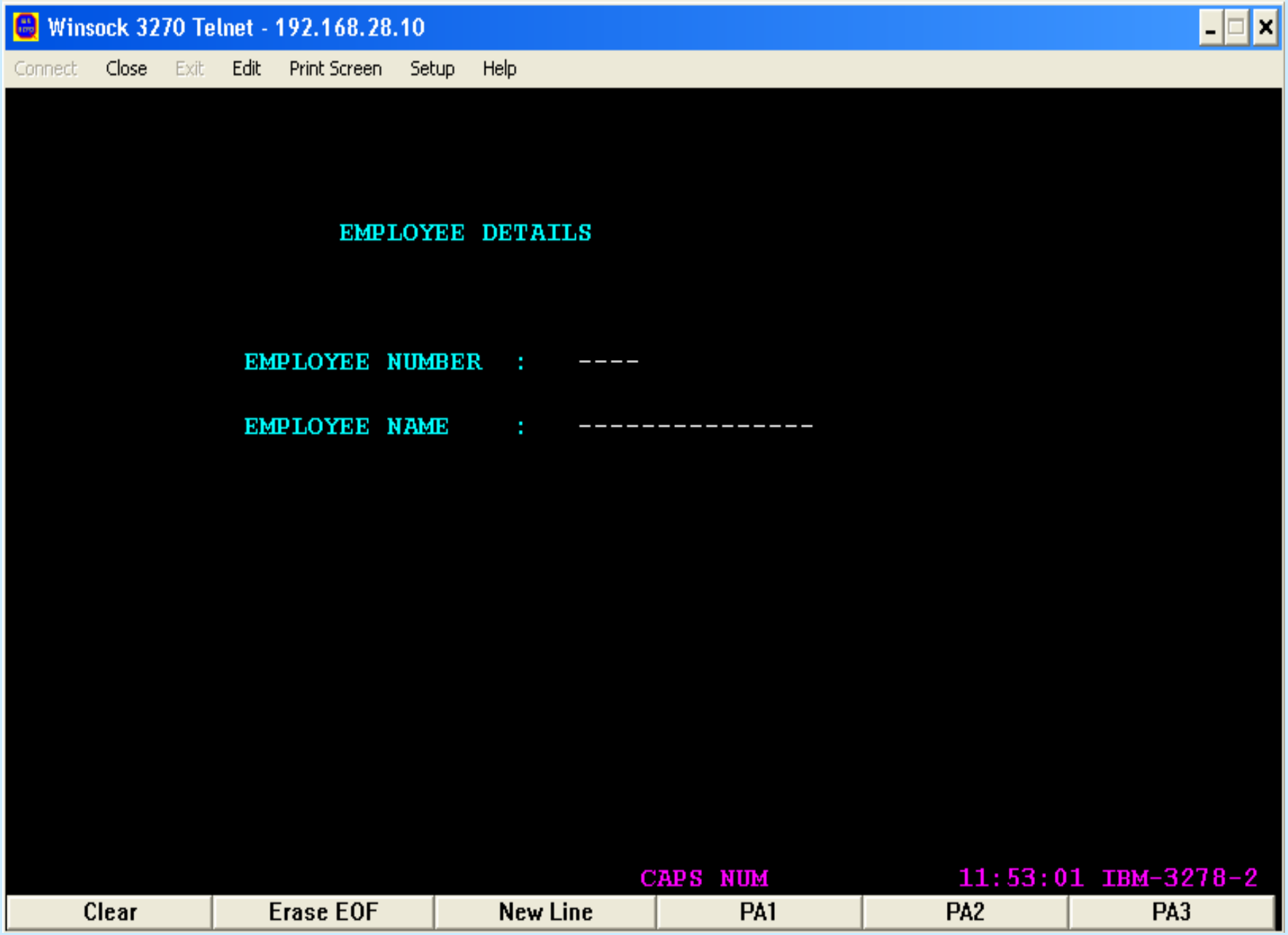
JCL to assemble macro coding

```
//MAPLE41C JOB NOTIFY=MAPLE41
//    JCLLIB  ORDER=(MAPLE41.CICS.TEMP)
//STEP01              EXEC  PROC=BMPJCL1,MEM=EMAP
//C.SYSIN          DD   DISP=SHR,DSN=MAPLE41.CICS.TEMP(&MEM)
//L.SYSLMOD DD DISP=SHR,DSN=CICSTS22.MAPLE.PRGLOAD3(MAPLE41
//CSYM.SYSIN   DD   DISP=SHR,DSN=MAPLE41.CICS.TEMP(&MEM)
//CSYM.SYSLIN DD   DISP=SHR,DSN=MAPLE41.SYSMB.PDS(&MEM)
```


Winsock 3270 Telnet - 192.168.28.10
Connect Close Exit Edit Print Screen Setup Help
SET PROG(MAPLE41) NE
STATUS: RESULTS - OVERTYPE TO MODIFY
Prog(MAPLE41) Leng(0000000344) Map Ena Pri NORMAL
Res(000) Use(0000000002) Qua
SYSID=CICS APPLID=A06C001
RESPONSE: NORMAL TIME: 11.52.32 DATE: 05.28.08
PF 1 HELP 3 END 5 VAR 7 SBH 8 SFH 9 MSG 10 SB 11 SF
CAPS NUM 11:58:27 IBM-3278-2
Clear
Erase EOF
New Line
PA1
PA2
PA3


Winsock 3270 Telnet - 192.168.28.10
Connect
Close
Exit
Edit
Print Screen
Setup
Help
CECI SEND MAP('EMP') MAPSET('MAPLE41')
CAPS NUM
11:59:36 IBM-3278-2
Clear
Erase EOF
New Line
PA1
PA2
PA3


Winsock 3270 Telnet - 192.168.28.10
Connect Close Exit Edit Print Screen Setup Help
SEND MAP('EMP') MAPSET('MAPLE41')
STATUS: COMMAND EXECUTION COMPLETE NAME=
EXEC CICS SENd Map('EMP ')
<< FROm() > < LEngth() > < DAtaonly > | MAPOnly >
< MAPSet('MAPLE41') >
< FMhparm() >
< Reqid() >
< LDc() | < ACTpartn() > < Outpartn() > >
< MSr() >
< Cursor() >
< Set() < MAPPingdev() > | PAging | Terminal < Wait > < LAst > >
< PRint >
< FREekb >
< ALArm >
< L40 | L64 | L80 | Honeom >
< NLeom >
< ERASE < DEfault | ALTernate > | ERASEAup >
< ACCum >
< FRSet >
+ < NOflush >
RESPONSE: NORMAL EIBRESP=+0000000000 EIBRESP2=+0000000000
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
CAPS NUM 11:57:18 IBM-3278-2
Clear Erase EOF New Line PA1 PA2 PA3


Winsock 3270 Telnet - 192.168.28.10
Connect
Close
Exit
Edit
Print Screen
Setup
Help
EMPLOYEE DETAILS
EMPLOYEE NUMBER : ----
EMPLOYEE NAME : ----------
CAPS NUM
11:53:01 IBM-3278-2
Clear
Erase EOF
New Line
PA1
PA2
PA3

Skipper Technique

```
000001 MAPLE41 DFHMSD TYPE=&SYSPARM,LANG=COBOL,STORAGE=AUTO,
X
 000002                          TIOAPFX=YES,MODE=INOUT
 000003 EMAP1     DFHMDI  SIZE=(24,80),LINE=1,COLUMN=1
 000004            DFHMDF POS=(5,10),ATTRB=(NORM,PROT),LENGTH=16,
X
 000005                          INITIAL='EMPLOYEE DETAILS'
 000006            DFHMDF POS=(7,7),ATTRB=(NORM,PROT),LENGTH=11,
X
 000007                          INITIAL='EMP NAME : '
 000008 ENAME    DFHMDF
POS=(7,20),ATTRB=(NORM,UNPROT,IC),LENGTH=10, X
 000009                          INITIAL='----------'
 000010            DFHMDF POS=(7,31),ATTRB=(NORM,ASKIP),LENGTH=1
 000011            DFHMDF POS=(9,7),ATTRB=(NORM,PROT),LENGTH=11,
X
 000012                          INITIAL='EMP ID   : '
 000013 ENUM      DFHMDF POS=(9,20),ATTRB=(NORM,UNPROT),LENGTH=5,
X
 000014                          INITIAL='-----'
 000015            DFHMSD TYPE=FINAL
```

Try Yourself!

- Design a map to get the student details like student ID, student name, marks for 3 major subjects.