1. Maps Complait status

EMSMS13 DFHMSD TYPE=&SYSPARM,MODE=INOUT,CTRL=FREEKB, X

LANG=COBOL,TIOAPFX=YES, X

DSATTS=(COLOR,HILIGHT), X

MAPATTS=(COLOR,HILIGHT), X

STORAGE=AUTO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CMRES DFHMDI SIZE=(24,80), X

LINE=1, X

COLUMN=1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(01,20), X

LENGTH=32, X

ATTRB=PROT, X

INITIAL='EDUCATION LOAN MANAGEMENT SYSTEM'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(02,20), X

LENGTH=32, X

ATTRB=PROT, X

INITIAL='--------------------------------'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(04,25), X

LENGTH=23, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='COMPLAINT STATUS WINDOW'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(08,08), X

LENGTH=10, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='USER ID :'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USRID DFHMDF POS=(08,20), X

LENGTH=05, X

ATTRB=(IC,UNPROT,FSET), X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(08,26), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(10,08), X

LENGTH=10, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='RESPONSE :'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

RESP DFHMDF POS=(10,20), X

LENGTH=30, X

ATTRB=PROT, X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(10,51), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(20,10), X

LENGTH=07, X

ATTRB=PROT, X

INITIAL='F3=BACK'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MSG1 DFHMDF POS=(22,02), X

LENGTH=67, X

ATTRB=PROT, X

COLOR=YELLOW

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMSD TYPE=FINAL

END

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DELETE FORM

EMSMS12 DFHMSD TYPE=&SYSPARM,MODE=INOUT,CTRL=FREEKB, X

LANG=COBOL,TIOAPFX=YES, X

DSATTS=(COLOR,HILIGHT), X

MAPATTS=(COLOR,HILIGHT), X

STORAGE=AUTO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USDEL DFHMDI SIZE=(24,80), X

LINE=1, X

COLUMN=1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(01,20), X

LENGTH=32, X

ATTRB=PROT, X

INITIAL='EDUCATION LOAN MANAGEMENT SYSTEM'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(02,20), X

LENGTH=32, X

ATTRB=PROT, X

INITIAL='--------------------------------'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(04,25), X

LENGTH=15, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='DELETE FORM '

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(08,08), X

LENGTH=10, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='USER ID :'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USRID DFHMDF POS=(08,20), X

LENGTH=05, X

ATTRB=(IC,UNPROT,FSET), X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(08,26), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(09,08), X

LENGTH=10, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='PASSWORD :'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PWD DFHMDF POS=(09,20), X

LENGTH=10, X

ATTRB=(IC,UNPROT,FSET), X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(09,31), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(10,08), X

LENGTH=10, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='MESSAGE :'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USRMSG DFHMDF POS=(10,20), X

LENGTH=40, X

ATTRB=(UNPROT,FSET), X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(10,60), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(20,10), X

LENGTH=07, X

ATTRB=PROT, X

INITIAL='F3=BACK'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MSG1 DFHMDF POS=(22,02), X

LENGTH=67, X

ATTRB=PROT, X

COLOR=YELLOW

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMSD TYPE=FINAL

END

DB2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Top of Data \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USREQ12 DFHMSD TYPE=&SYSPARM,MODE=INOUT,CTRL=FREEKB, X

LANG=COBOL,TIOAPFX=YES, X

DSATTS=(COLOR,HILIGHT), X

MAPATTS=(COLOR,HILIGHT), X

STORAGE=AUTO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USRDB2 DFHMDI SIZE=(24,80), X

LINE=1, X

COLUMN=1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(01,18), X

LENGTH=17, X

ATTRB=PROT, X

INITIAL='DB2 LOGIN SCREEN'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(02,18), X

LENGTH=17, X

ATTRB=PROT, X

INITIAL='================='

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(04,15), X

LENGTH=13, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='USER ID :'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USRID DFHMDF POS=(04,31), X

LENGTH=06, X

ATTRB=(IC,UNPROT,FSET), X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(04,38), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(06,15), X

LENGTH=13, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='PASSWORD :'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PWD DFHMDF POS=(06,31), X

LENGTH=06, X

ATTRB=(DRK,UNPROT,FSET), X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(06,38), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(15,10), X

LENGTH=07, X

ATTRB=PROT, X

INITIAL='F3=EXIT'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(20,15), X

LENGTH=21, X

ATTRB=PROT, X

COLOR=TURQUOISE, X

INITIAL='ENQUIRY USER ID: '

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EUSRID DFHMDF POS=(20,31), X

LENGTH=06, X

ATTRB=(IC,UNPROT,FSET), X

HILIGHT=UNDERLINE, X

COLOR=WHITE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMDF POS=(20,38), X

LENGTH=01, X

ATTRB=ASKIP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MSG1 DFHMDF POS=(23,02), X

LENGTH=67, X

ATTRB=PROT, X

COLOR=YELLOW

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DFHMSD TYPE=FINAL

END

maps

       IDENTIFICATION DIVISION.

       PROGRAM-ID. MPUSEQ1.

       ENVIRONMENT DIVISION.

       DATA DIVISION.

       WORKING-STORAGE SECTION.

       01 WS-COMMAREA          PIC X(10).

            COPY USREQ12.

            COPY DFHAID.

            COPY DFHBMSCA.

       LINKAGE SECTION.

       01 DFHCOMMAREA          PIC X(10).

       PROCEDURE DIVISION.

            IF EIBCALEN > ZERO

              MOVE DFHCOMMAREA TO WS-COMMAREA

           END-IF

           EVALUATE TRUE

             WHEN EIBCALEN = ZERO

               MOVE LOW-VALUES TO USRENQO

               PERFORM SEND-MAP

             WHEN EIBAID = DFHCLEAR

               MOVE LOW-VALUES TO USRENQO

               PERFORM SEND-MAP

             WHEN EIBAID = DFHPA1 OR DFHPA2 OR DFHPA3

               CONTINUE

             WHEN EIBAID = DFHPF3 OR DFHPF12

               MOVE LOW-VALUES TO USRENQO

               MOVE 'LEAVING SCREEN -PRESS CLEAR KEY!' TO MSG1O

                            PERFORM SEND-MAP

               EXEC CICS

                RETURN

               END-EXEC

             WHEN EIBAID = DFHENTER

               PERFORM PROCESS-PARA

             WHEN OTHER

               MOVE LOW-VALUES TO USRENQO

               MOVE 'INVALID KEY PRESSED' TO MSG1O

               PERFORM SEND-MAP-DATA

           END-EVALUATE.

            EXEC CICS

              RETURN TRANSID('P124')

              COMMREA (WS-COMMAREA)

            END-EXEC.

      PROCESS-PARA.

           PERFORM RECEIVE-MAP.

           IF PWDO = 'ABCDEF'

              MOVE 'USER LOGIN SUCCESSFULL' TO MSG1O

           ELSE

             MOVE 'USER LOGIN NOT SUCCESSFULL' TO MSG1O

           END-IF

          MOVE -1 TO USRIDL.

           PERFORM SEND-MAP-DATA.

           EXIT.

        SEND-MAP.

             EXEC CICS SEND

                MAP    ('USRENQ')

                MAPSET ('USRENQ12')

                FROM   (USRENQO)

                ERASE

             END-EXEC.

           EXIT.

        SEND-MAP-DATA.

             EXEC CICS SEND

                MAP    ('USRENQ')

                MAPSET ('USRENQ12')

                FROM   (USRENQO)

                CURSOR

                DATAONLY

             END-EXEC.

        RECEIVE-MAP.

            EXEC CICS RECEIVE

               MAP    ('USRENQ')

               MAPSET ('USRENQ12')

               INTO   (USRENQI)

            END-EXEC.

Eval

PROCEDURE DIVISION.

      IF EIBCALEN = ZERO

        MOVE DFHCOMMAREA TO WS-COMMAREA

      END-IF

      EVALUATE TRUE

        WHEN EIBCALEN = ZERO

          MOVE LOW-VALUES TO USRENQO

          PERFORM SEND-MAP

        WHEN EIBAID = DFHCLEAR

          MOVE LOW-VALUES TO USRENQO

          PERFORM SEND-MAP

        WHEN EIBAID = DFHPA1 OR DFHPA2 OR DFHPA3

          CONTINUE

         WHEN EIBAID = DFHPF3 OR DFHPF12

           MOVE LOW-VALUES TO USRENQO

           MOVE 'LEAVING SCREEN -PRESS CLEAR KEY!' TO MSG1O

           PERFORM SEND-MAP

           EXEC CICS

            RETURN

           END-EXEC

         WHEN EIBAID = DFHENTER

           PERFORM PROCESS-PARA

         WHEN OTHER

           MOVE LOW-VALUES TO USRENQO

           MOVE 'INVALID KEY PRESSED' TO MSG1O

           PERFORM SEND-MAP-DATA

END-PERFORM.

       IDENTIFICATION DIVISION.

       PROGRAM-ID. CICSPG01.

       ENVIRONMENT DIVISION.

       DATA DIVISION.

       WORKING-STORAGE SECTION.

       01 WS-INPUT.

          05 WS-INTEXT     PIC X(15).

          05 WS-MESSAGE-I  PIC X(85) VALUE SPACES.

       01 WS-OUTPUT-1.

          05 WS-MESSAGE-1  PIC X(100) VALUE SPACES.

       01 WS-OUTPUT-2.

          05 WS-MESSAGE-2  PIC X(100) VALUE SPACES.

       01 WS-MSG-LENGTH    PIC 9(04) COMP.

       01 WS-COMM          PIC X(03).

       COPY DFHAID.

       LINKAGE SECTION.

       01 DFHCOMMAREA.

          05 DFH-VAR       PIC X(03).

       PROCEDURE DIVISION.

       100-PARA.

            MOVE 100  TO  WS-MSG-LENGTH.

            MOVE 'WELCOME TO CICS WORLD - PRESS ENTER KEY NOW !!!'

                 TO WS-OUTPUT-1

            EXEC CICS SEND

                LENGTH(WS-MSG-LENGTH)

                ERASE

           END-EXEC.

           MOVE 100  TO  WS-MSG-LENGTH.

           EXEC CICS RECEIVE

               INTO (WS-INTEXT)

               LENGTH (WS-MSG-LENGTH)

           END-EXEC.

           MOVE 'THIS IS A TEST PROGRAM FOR CICS!!!'

                TO WS-OUTPUT-1

           EXEC CICS SEND

                FROM(WS-OUTPUT-1)

                LENGTH(WS-MSG-LENGTH)

                ERASE

           END-EXEC.

           EXEC CICS RETURN END-EXEC.

COMMANDS………………………………………………..

CEDA DEF PROG(CICSPXXX) GROUP(OZAXXX)

CEDA I PROG(CICSPXXX) GROUP(OZAXXX)

CEDA DEF TRANSACTION(PXXX) GROUP(OZAXXX)

CEDA I TRANSACTION(PXXX) GROUP(OZAXXX)

Db2

IDENTIFICATION DIVISION.

       PROGRAM-ID. DB2HO01.

       ENVIRONMENT DIVISION.

       CONFIGURATION SECTION.

       SOURCE-COMPUTER. DELL.

       OBJECT-COMPUTER. DELL.

       INPUT-OUTPUT SECTION.

       FILE-CONTROL.

            SELECT INFILE ASSIGN TO INFILE.

       DATA DIVISION.

       FILE SECTION.

       FD INFILE.

       01 IN-REC       PIC X(50).

       WORKING-STORAGE SECTION.

       01 WS-IN            PIC X(06).

       01 WS-EOF           PIC X.

       01 WS-SQLCODE       PIC +9(09).

       01 WS-REC.

          05 WS-ID         PIC X(06).

          05 WS-NAME       PIC X(12).

          05 WS-DEPT       PIC X(06).

          05 WS-SALARY     PIC 9(05)V99.

          05 WS-DATA       PIC 9(49).

             EXEC SQL

                  INCLUDE EMPLOYEE

             END-EXEC

             EXEC SQL

                  INCLUDE SQLCA

             END-EXEC

       PROCEDURE DIVISION.

       100-PARA.

           OPEN INPUT INFILE.

           PERFORM 100 READ-PARA UNTIL WS-EOF = 'Y'

           PERFORM 200-PARA.

           CLOSE INFILE.

           STOP RUN.

       100-READ PARA.

           READ INFILE INTO WS-REC AT END MOVE 'Y' TO WS-EOF

           NOT AT END

           MOVE WS-SALARY TO SALARY OF DCLEMPLOYEE

           EXEC SQL

                INSERT INTO EMPLOYEE

                VALUES (:WS-ID,

                        :WS-NAME,

                        :WS-DEPT,

                        :WS-SALARY)

           END-EXEC

           EVALUTE SQLCODE

               WHEN 0

                    DISPLAY 'ROW  INSERTED'

               WHEN OTHER

                    MOVE SQLCODE TO WS-SQLCODE

                    DISPLAY 'SQL CODE IS: ' WS-SQLCODE

                    DISPLAY ' NOT INSERTED'

                    GO TO 200-PARA

           END-EVALUATE

     200-PARA.

           ACCEPT WS-IN

           IF WS-IN = 'XXX'

              DISPLAY ' END OF DATA'

           END-IF

           EXEC SQL

               SELECT EMP ID,

                      EMP NAME

               INTO   : EMP-ID

                      : EMP-NAME

              FROM EMPLOYEE

              WHERE EMP\_ID = WS-IN

          END-EXEC

          EVALUTE SQLCODE

              WHEN 0

                   DISPLAY 'EMP-ID: ' WS-IN

                   DISPLAY 'EMP-NAME: ' WS-NAME

              WHEN 100

                   MOVE SQLCODE TO WS-SQLCODE

                   DISPLAY 'SQL CODE IS: ' WS-SQLCODE

                   DISPLAY 'ID MATCH NOT FOUND:' WS-IN

              WHEN OTHER

                   DISPLAY 'MATCH FOUND FOR ID: ' WS-IN

          END-EVALUATE

vsam

     IDENTIFICATION DIVISION.

       PROGRAM-ID. VSAMHO4.

       ENVIRONMENT DIVISION.

       INPUT-OUTPUT SECTION.

       FILE-CONTROL.

            SELECT EMPFILE ASSIGN TO EMPFILE

                   ORGANIZATION IS INDEXED

                   ACCESS MODE IS DYNAMIC

                   RECORD KEY IS EMP-ID

                   ALTERNATE RECORD KEY IS EMP-DEPT

                   FILE STATUS IS STAT1.

       DATA DIVISION.

       FILE SECTION.

       FD EMPFILE.

       01 EMP-REC.

          05 EMP-ID    PIC X(05).

          05 EMP-NAME  PIC X(10).

        05 EMP-DEPT  PIC X(05).

        05 EMP-LOC   PIC X(05).

        05 EMP-DATA  PIC X(25).

     WORKING-STORAGE SECTION.

     01 WS-INPUT      PIC X(05).

     01 WS-EOF        PIC X.

     01 STAT1         PIC X(02).

     01 WS-DEPT       PIC X(05).

     PROCEDURE DIVISION.

     100-PARA.

         OPEN I-O EMPFILE.

         DISPLAY ' STATUS AFTER OPENING THE FILE: ' STAT1

         ACCEPT WS-INPUT

         PERFORM 300-PARA.

            CLOSE EMPFILE.

            STOP RUN.

      300-PARA.

          MOVE WS-INPUT TO EMP-ID

          READ EMPFILE

          DISPLAY ' STATUS AFTER READING 1 THE FILE: ' STAT1

          MOVE EMP-DEPT TO WS-DEPT

          START EMPFILE KEY = EMP-DEPT

          DISPLAY ' STATUS AFTER STARTING THE FILE: ' STAT1

          PERFORM UNTIL WS-EOF = 'Y'

             READ EMPFILE NEXT

             DISPLAY ' EMP-ID  : ' EMP-ID

             DISPLAY ' EMP-LOC : '   EMP-LOC

             DISPLAY ' STATUS AFTER READING 2 THE FILE: ' STAT1

             IF STAT1 = '10'

                MOVE 'Y' TO WS-EOF

             END-IF

             IF EMP-DEPT = WS-DEPT

                MOVE 'LAAAA' TO EMP-LOC

                REWRITE EMP-REC

             END-IF

          END-PERFORM.

VSAM AIX

AIX PROGRAMMING

===============

COBOL PROGRAM

=============

FILE-CONTROL.

SELECT EMPFILE ASSIGN TO EMPFILE

ORGANIZATION IS INDEXED

ACCESS MODE IS DYNAMIC

RECORD KEY IS EMP-ID

ALTERNATE RECORD KEY IS EMP-DEPT

FILE STATUS IS STAT1.

START EMPFILE KEY = EMP-DEPT

DISPLAY 'FILE STATUS CODE AFTER START : ' STAT1

PERFORM UNTIL WS-EOF = 'Y'

READ EMPFILE NEXT

DISPLAY 'EMP ID IS : ' EMP-ID

DISPLAY 'EMP LOC IS : ' EMP-LOC

DISPLAY 'FILE STATUS AFTER READ IS : ' STAT1

RUN JCL

=======

//OZA101A JOB (123),'MFT',CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),

// NOTIFY=&SYSUID,REGION=0M

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* JCL JOB EXECUTING THE IEFBR14 UTILTITY TO ALLOCATE PS DATA SE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//STEP10 EXEC PGM=VSAMPGM4

//STEPLIB DD DSN=OZA101.MFT.LOADLIB,DISP=SHR

//SYSPRINT DD SYSOUT=\*

//SYSOUT DD SYSOUT=\*

//SYSDUMP DD SYSOUT=\*

//EMPFILE DD DSN=OZA101.MFT.KSDSTEST,DISP=SHR

//EMPFILE1 DD DSN=OZA101.MFT.PATHEMPL,DISP=SHR

//SYSIN DD \*

A1112

/\*

Dynamic

      IDENTIFICATION DIVISION.

      PROGRAM-ID. VSAMHO03.

      ENVIRONMENT DIVISION.

      INPUT-OUTPUT SECTION.

      FILE-CONTROL.

           SELECT KSDSFILE ASSIGN TO KSDSFILE

                  ORGANIZATION IS INDEXED

                  ACCESS MODE IS DYNAMIC

                  RECORD KEY IS KSDS-ID

                  FILE STATUS IS STAT1.

      DATA DIVISION.

      FILE SECTION.

      FD KSDSFILE.

      01 KSDS-REC.

         05 KSDS-ID    PIC X(05).

         05 KSDS-DATA  PIC X(45).

       WORKING-STORAGE SECTION.

       01 WS-FILLER-Z   PIC X(45) VALUE ALL 'Z'.

       01 WS-INPUT.

          05 WS-IN1     PIC X(05).

          05 WS-IN2     PIC X(05).

       01 STAT1         PIC X(02).

       PROCEDURE DIVISION.

       100-PARA.

           ACCEPT WS-INPUT

           IF WS-INPUT = '----'

              DISPLAY 'END OF INPUT'

              STOP RUN

           END-IF

           OPEN I-O KSDSFILE

           DISPLAY 'FILE STATUS AFTER OPEN IS : ' STAT1

           MOVE WS-IN1 TO KSDS-ID

           START KSDSFILE KEY >= KSDS-ID

           PERFORM UNTIL KSDS-ID = WS-IN2

              READ KSDSFILE NEXT

              AT END CLOSE KSDSFILE

              NOT AT END

                 MOVE WS-FILLER-Z TO KSDS-DATA

                 REWRITE KSDS-REC

              END-READ

           END-PERFORM

           STOP RUN.

Vsam

        IDENTIFICATION DIVISION.

        PROGRAM-ID. VSAMPGM1.

        ENVIRONMENT DIVISION.

        INPUT-OUTPUT SECTION.

        FILE-CONTROL.

            SELECT INFILE  ASSIGN TO INF.

            SELECT OUTFILE ASSIGN TO KSDSFILE

            ORGANIZATION IS INDEXED

            ACCESS MODE IS RANDOM

            RECORD KEY IS OUT-DATA-ID

            FILE STATUS IS FSTATUS.

        DATA DIVISION.

        FILE SECTION.

        FD INFILE.

        01 IN-REC.

           10 IN-DATA-ID  PIC X(05).

           10 IN-NAME     PIC X(45).

       FD OUTFILE.

       01 OUT-REC.

          10 OUT-DATA-ID  PIC X(05).

          10 OUT-NAME     PIC X(45).

       WORKING-STORAGE SECTION.

       01 WS-EOF           PIC X.

       01 FSTATUS          PIC 9(02).

       01 WS-INPUT.

          05 WS-DATA       PIC X(5).

          05 WS-OPRN       PIC X(1).

       PROCEDURE DIVISION.

       200-PARA.

           ACCEPT WS-INPUT

           IF WS-INPUT = 'XXX'

              DISPLAY "FILE PROCESSING OVER"

              CLOSE OUTFILE

              STOP RUN

          END-IF

          OPEN I-O OUTFILE

          MOVE WS-DATA TO OUT-DATA-ID

          READ  OUTFILE

          DISPLAY "STATUS         " FSTATUS WS-OPRN

          EVALUATE TRUE

                  WHEN FSTATUS =  00  AND WS-OPRN = 'I'

                      DISPLAY  "DATA IS PRESENT CANT INSERT"

                  WHEN FSTATUS =  23  AND WS-OPRN = 'I'

                     MOVE WS-DATA TO OUT-DATA-ID

                     MOVE "IIIIIIII" TO OUT-NAME

                     WRITE OUT-REC

                     DISPLAY  "INSERTED SUCCESFULLY"

                  WHEN FSTATUS =  00  AND WS-OPRN = 'D'

                     MOVE WS-DATA TO OUT-DATA-ID

                     DELETE OUTFILE

                     DISPLAY  "DELETED SUCCESSFULLY      "

                   WHEN FSTATUS =  23  AND WS-OPRN = 'D'

                      DISPLAY  "NOT PRESENT CANT DELETE"

                   WHEN FSTATUS =  00  AND WS-OPRN = 'U'

                      MOVE WS-DATA TO OUT-DATA-ID

                      MOVE "YYYYY" TO OUT-NAME

                      REWRITE OUT-REC

                      DISPLAY  "UPDATED SUCCESSFULLY      "

                   WHEN FSTATUS =  23  AND WS-OPRN = 'U'

                      DISPLAY  "NOT PRESENT CANT UPDATE"

               END-EVALUATE

           GO TO 200-PARA.

----------==================================================

Cob02

IDENTIFICATION DIVISION.

PROGRAM-ID. USERINF2.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT INFILE ASSIGN TO INDD.

SELECT TEMPFLE ASSIGN TO TEMP1.

DATA DIVISION.

FILE SECTION.

FD INFILE.

01 IN-REC.

05 IN-ID PIC X(05).

05 IN-NAME PIC X(10).

05 IN-DEPT PIC X(05).

05 IN-LOC PIC X(30).

FD TEMPFLE.

01 TEMP-ID.

05 IN-TEMP PIC X(05).

05 IN-BLNK PIC X(45).

WORKING-STORAGE SECTION.

01 WS-EOF PIC X.

01 WS-REC PIC X(50).

01 WS-INPUT PIC X(05).

01 WS-COUNT PIC 9(2) VALUE 0.

01 WS-TOTAL PIC 9(2) VALUE 0.

01 WS-MATCH PIC 9(2) VALUE 0.

01 WS-UNMATCH PIC 9(2) VALUE 0.

01 WS-OCCFLG PIC 9(2) VALUE 1.

PROCEDURE DIVISION.

100-PARA.

ACCEPT WS-INPUT

IF WS-INPUT = 'XXX'

DISPLAY 'TOTAL RECORD : ' WS-TOTAL

SUBTRACT WS-UNMATCH FROM WS-TOTAL GIVING WS-MATCH

DISPLAY 'TOTAL MATCH FOUND : ' WS-MATCH

DISPLAY 'TOTAL MISSING : ' WS-UNMATCH

DISPLAY 'NEWELY ADDED : ' WS-UNMATCH

STOP RUN

END-IF

ADD 1 TO WS-TOTAL

OPEN INPUT TEMPFLE

MOVE 0 TO WS-OCCFLG

PERFORM 300-READ-PARA UNTIL WS-EOF = 'Y'

MOVE 'N' TO WS-EOF

IF WS-OCCFLG NOT = 0

DISPLAY WS-INPUT " ALREDY PROCESSED (" WS-OCCFLG ")"

END-IF

CLOSE TEMPFLE

OPEN EXTEND TEMPFLE

MOVE WS-INPUT TO IN-TEMP

WRITE TEMP-ID

CLOSE TEMPFLE

OPEN INPUT INFILE

PERFORM 200-READ-PARA UNTIL WS-EOF = 'Y'

CLOSE INFILE

OPEN EXTEND INFILE

IF WS-COUNT = 0

ADD 1 TO WS-UNMATCH

DISPLAY 'MATCH NOT FOUND FOR ' WS-INPUT

MOVE WS-INPUT TO IN-ID

MOVE "YYYYYYYYYY" TO IN-NAME

MOVE "YYYYY" TO IN-DEPT

MOVE "YYYYYY" TO IN-LOC

DISPLAY 'BUT ADDED WITH DEFAULT VALUES'

WRITE IN-REC

END-IF

DISPLAY '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

MOVE 'N' TO WS-EOF

MOVE 0 TO WS-COUNT

CLOSE INFILE

GO TO 100-PARA.

200-READ-PARA.

READ INFILE AT END MOVE 'Y' TO WS-EOF

NOT AT END

IF IN-ID = WS-INPUT

ADD 1 TO WS-COUNT

DISPLAY 'MATCH FOUND FOR ' WS-INPUT "(" WS-COUNT ")"

END-IF

END-READ.

300-READ-PARA.

READ TEMPFLE AT END MOVE 'Y' TO WS-EOF

NOT AT END

IF IN-TEMP = WS-INPUT

ADD 1 TO WS-OCCFLG

END-IF

END-READ.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Top of Data \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//OZA117A JOB (123),'MFT',CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),

// NOTIFY=&SYSUID,REGION=6M

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* JCL JOB EXECUTING THE IEFBR14 UTILTITY TO ALLOCATE PS DATA SET

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//STEP10 EXEC PGM=IEFBR14

//DD1 DD DSN=OZA117.MFT.TEST.OUTFILE8,

// DISP=(MOD,DELETE,DELETE),

// SPACE=(TRK,(10,10),RLSE)

//STEP11 EXEC PGM=IEFBR14

//DD1 DD DSN=OZA117.MFT.TEST.TEMP,

// DISP=(MOD,DELETE,DELETE),

// SPACE=(TRK,(10,10),RLSE)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* JCL JOB EXECUTING THE IEFBR14 UTILTITY TO ALLOCATE PS DATA SET

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//STEP20 EXEC PGM=USERINF2

//STEPLIB DD DSN=OZA117.MFT.LOADLIB,DISP=SHR

//SYSOUT DD SYSOUT=\*

//INDD DD DSN=OZA117.MFT.TEST.PSFILE9,DISP=SHR

//OUTDD DD DSN=OZA117.MFT.TEST.OUTFILE8,

// DISP=(NEW,CATLG,DELETE),

// SPACE=(TRK,(10,10),RLSE),

// UNIT=SYSDA,

// DCB=(RECFM=FB,LRECL=50,BLKSIZE=500)

//TEMP1 DD DSN=OZA117.MFT.TEST.TEMP,

// DISP=(NEW,CATLG,DELETE),

// SPACE=(TRK,(10,10),RLSE),

// UNIT=SYSDA,

// DCB=(RECFM=FB,LRECL=50,BLKSIZE=500)

//SYSIN DD \*

A1111

B1111

M1111

A1111