**Lab objectives:**

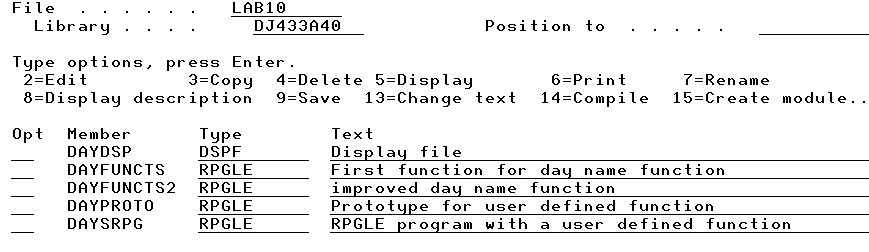
**- Create and Use user defined functions in a RPGLE program**

**- Note Lab 10 has been simplified to allow you time to work on Lab 8**

**Requirements to pass the lab:**

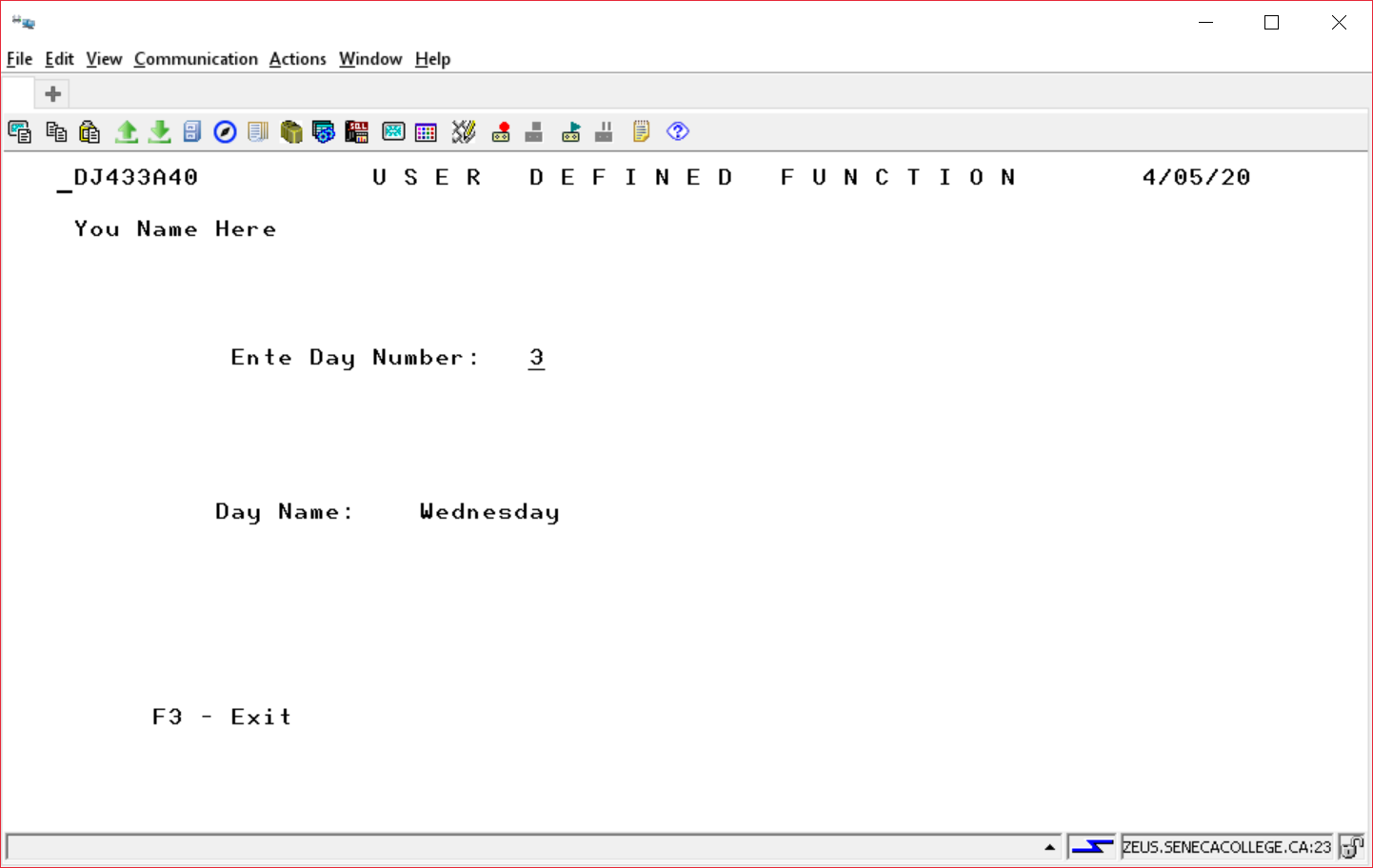
Successfully run the RUNLAB10 CLLE driver program demonstrating user defined functions implemented in the WHATDAY program.

User Defined Functions

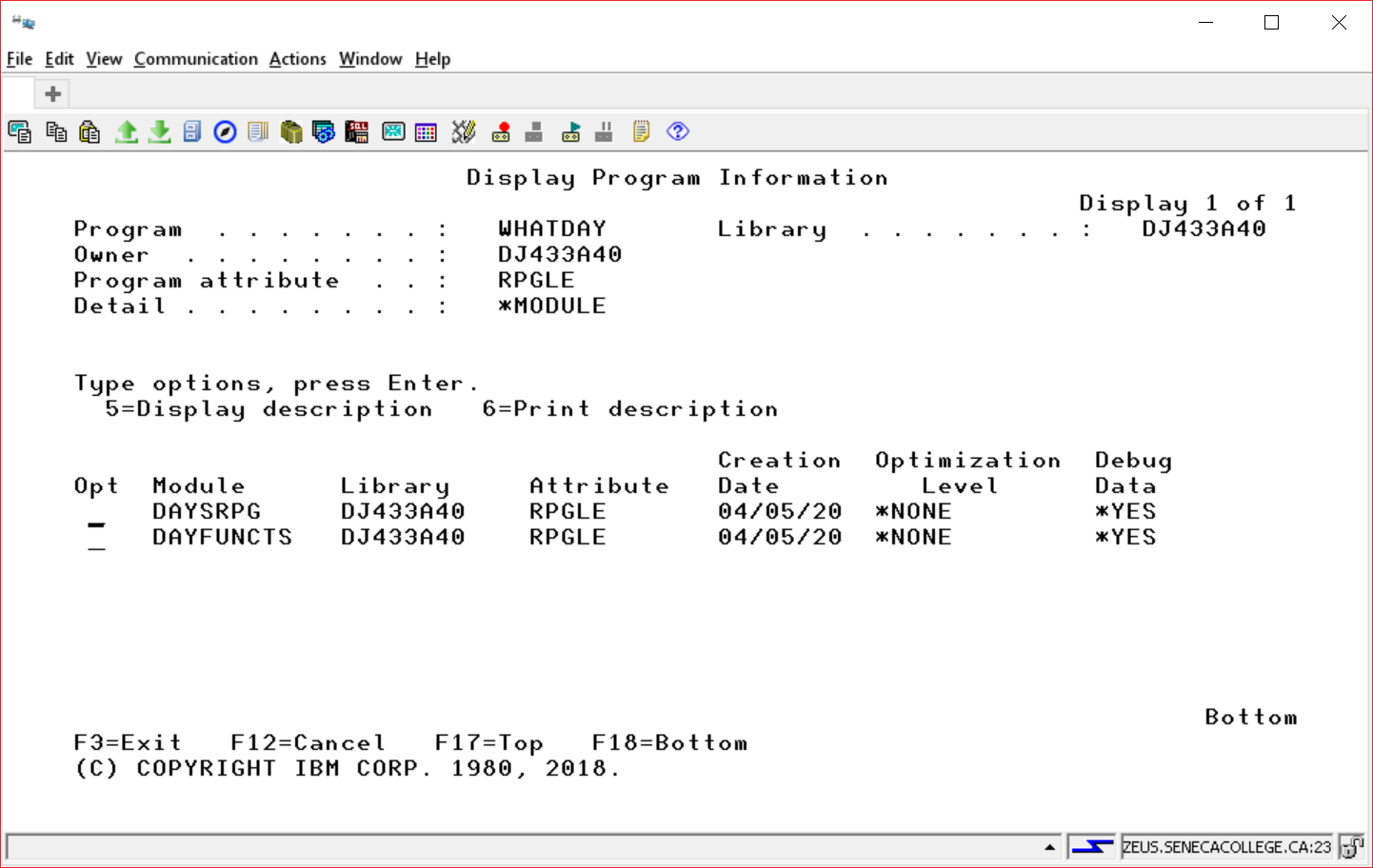


A user defined function is created and used. DayNumName( ) accepts a one digit number and returns a day name word. First this user defined function is set up using inline case structure as a simple solution and that code is entered in DAYFUNCTS. A better solution using an array will be discussed in class and will be entered in DAYFUNCTS2.

Input and Output records for DAYSRPG which relies on a user defined function



The module that relies on a user defined function is combined with the module that solves the user defined function into a working program called WHATDAY



Partial code for DAYSRPG, DAYFUNCTS and RUNLAB10 is available in BCI433LIB/LAB10W2020. All of the code for DAYDSP is available at that location. None of the code for DAYFUNCTS2 and DAYPROTO is available to copy, that code will be provided in class.

DAYSRPG

DCL-f DayDsp Workstn;

/COPY LAB10INTRO,DAYPROTO

EXFMT INPUT;

DOW NOT(\*IN03);

DayName = DayNumName(DayIn);

\*in99 = \*on;

WRITE INPUT;

EXFMT OUTPUT;

\*in99 = \*off;

IF \*IN03 = '0';

DayIn = 0;

EXFMT INPUT ;

ENDIF;

ENDDO;

\*INLR = \*ON;

RETURN;

DAYPROTO contains two prototypes

The Prototype for DayNumName

Dcl-Pr DayNumName Char(9);

DayIn Packed(1);

End-Pr;

The Prototype for MonthNumName (not used in this lab)

Used to support a more sophisticated Day Name like

Sunday April 5, 2020

Dcl-Pr MonthNumName Char(9);

DayIn Packed(2);

End-Pr;

DayNumName and MonthNumName are not RPGLE functions. But, they can be created and then used by all programmers. The RPGLE compiler would reject

DayName = DayNumName(DayIn);

The prototype is telling the compiler to accept DayNumName as legitimate.

DAYFUNCTS

Ctl-Opt NoMain; (no need for \*INLR = \*ON)

/COPY LAB10INTRO, DAYPROTO (Get the Prototype code)

Dcl-Proc DayNumName EXPORT; ( the user defined function)

Dcl-pi \*N CHAR(9); (what is returned from the function)

Number Packed(1); (what is input to the function)

End-PI;

DCL-s DayName Char(9); (Local variable)

(Solve the function)

SELECT;

WHEN NUMBER = 1;

DAYNAME = 'Monday';

WHEN NUMBER = 2;

DAYNAME = 'Tuesday';

WHEN NUMBER = 3;

DAYNAME = 'Wednesday';

WHEN NUMBER = 4;

DAYNAME = 'Thursday';

WHEN NUMBER = 5;

DAYNAME = 'FRIDAY';

WHEN NUMBER = 6;

DAYNAME = 'Saturday';

WHEN NUMBER = 7;

DAYNAME = 'Sunday';

OTHER;

DAYNAME = 'Unknown';

ENDSL;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

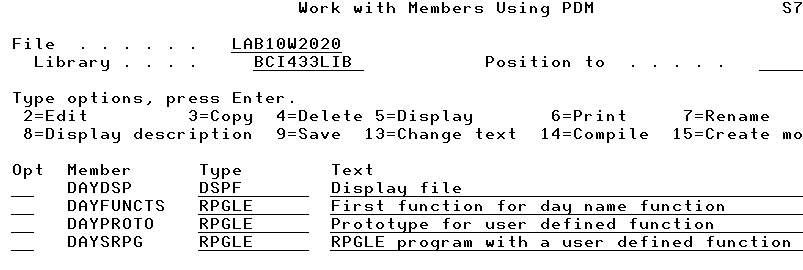
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DAYFUNCTS2

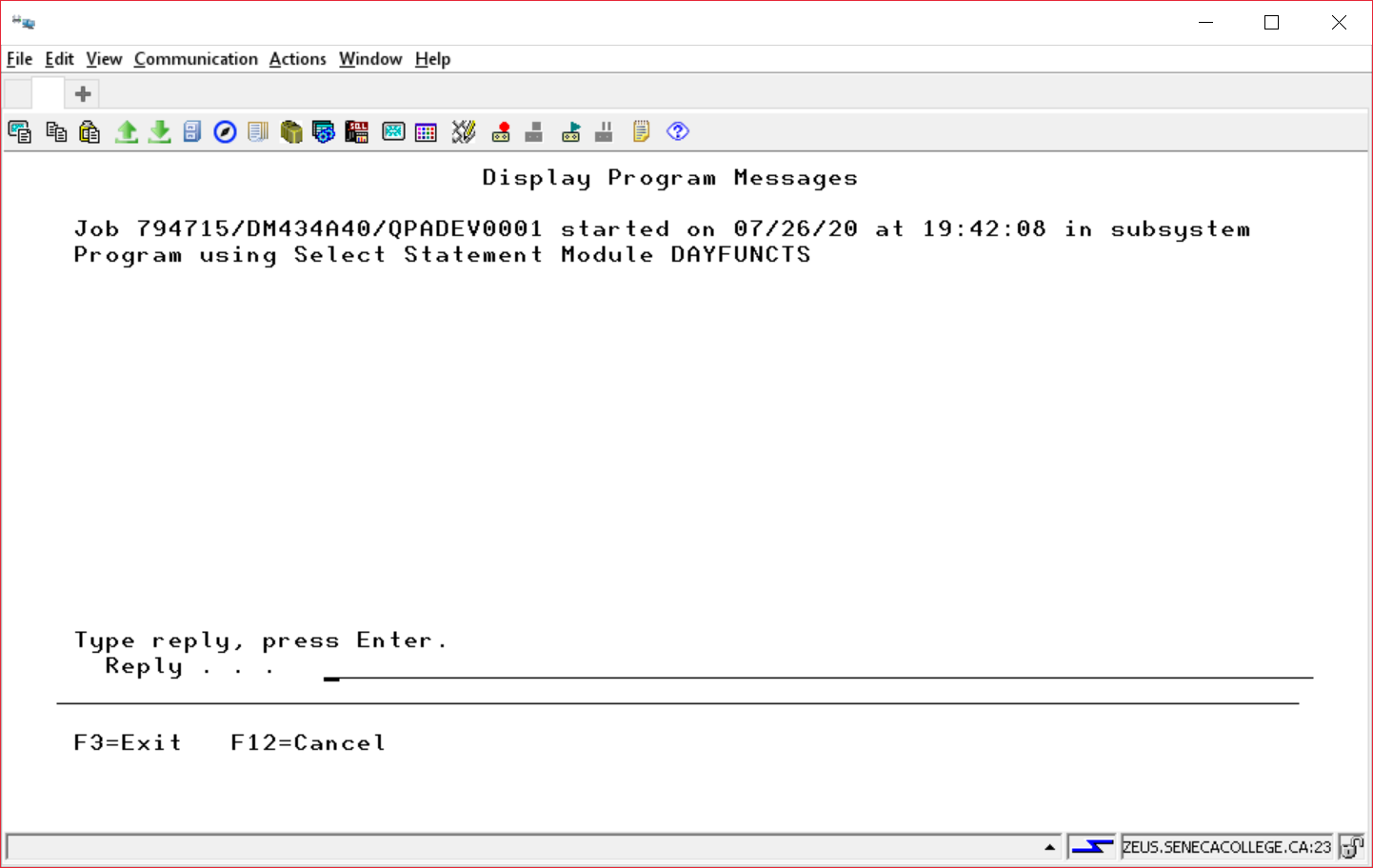
This program will be discussed in class. It will replace the in line case structure with a positional array lookup. It will not error check for an invalid number like the solution above did. In the lab 11, it will be impossible to send the day function a weekday number that did not exist (like 9).

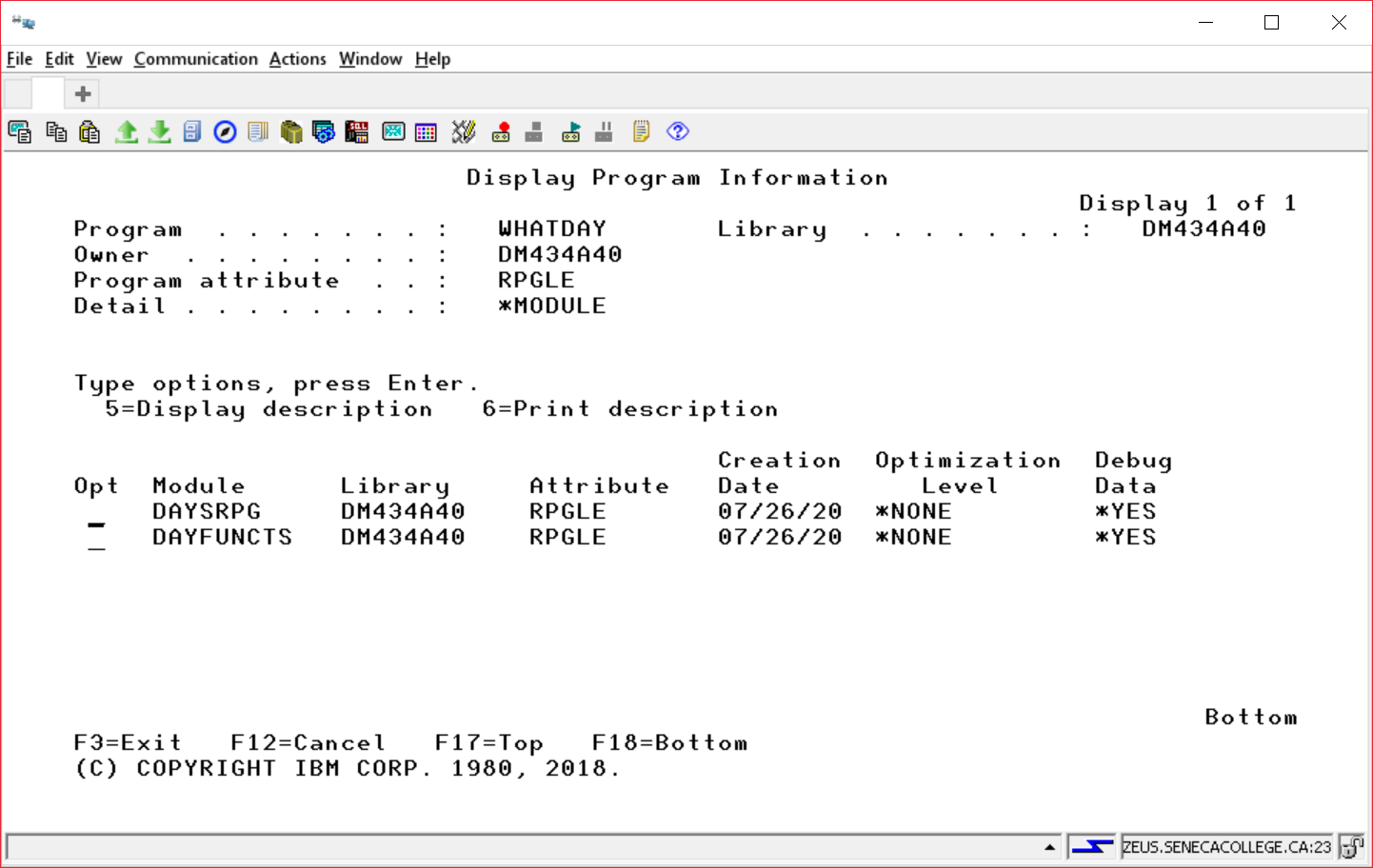
RUNLAB10

The following code is available for you to copy.

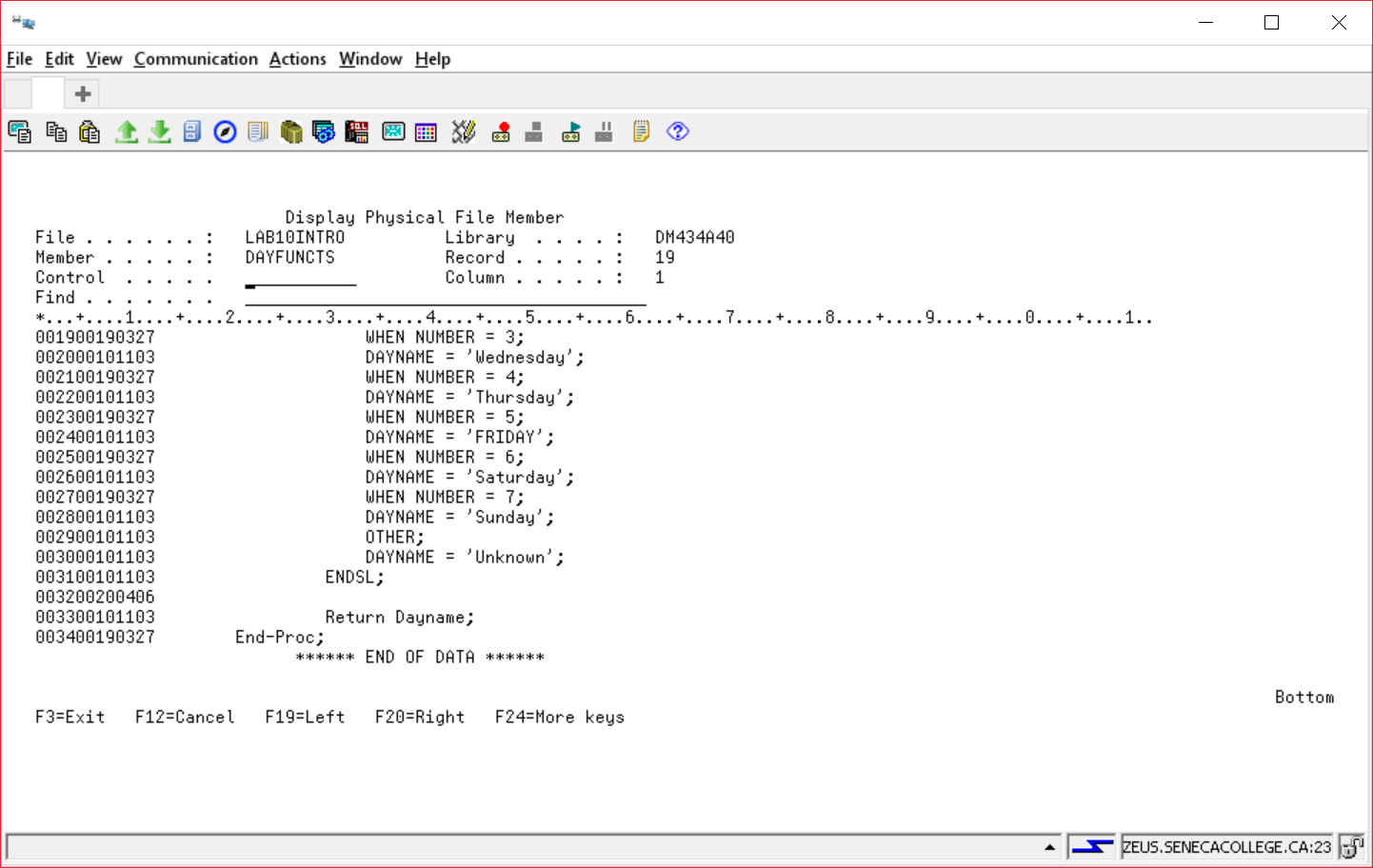


You need to code a CLLE driver program discussed in class to demonstrate this lab. The following screens will show.

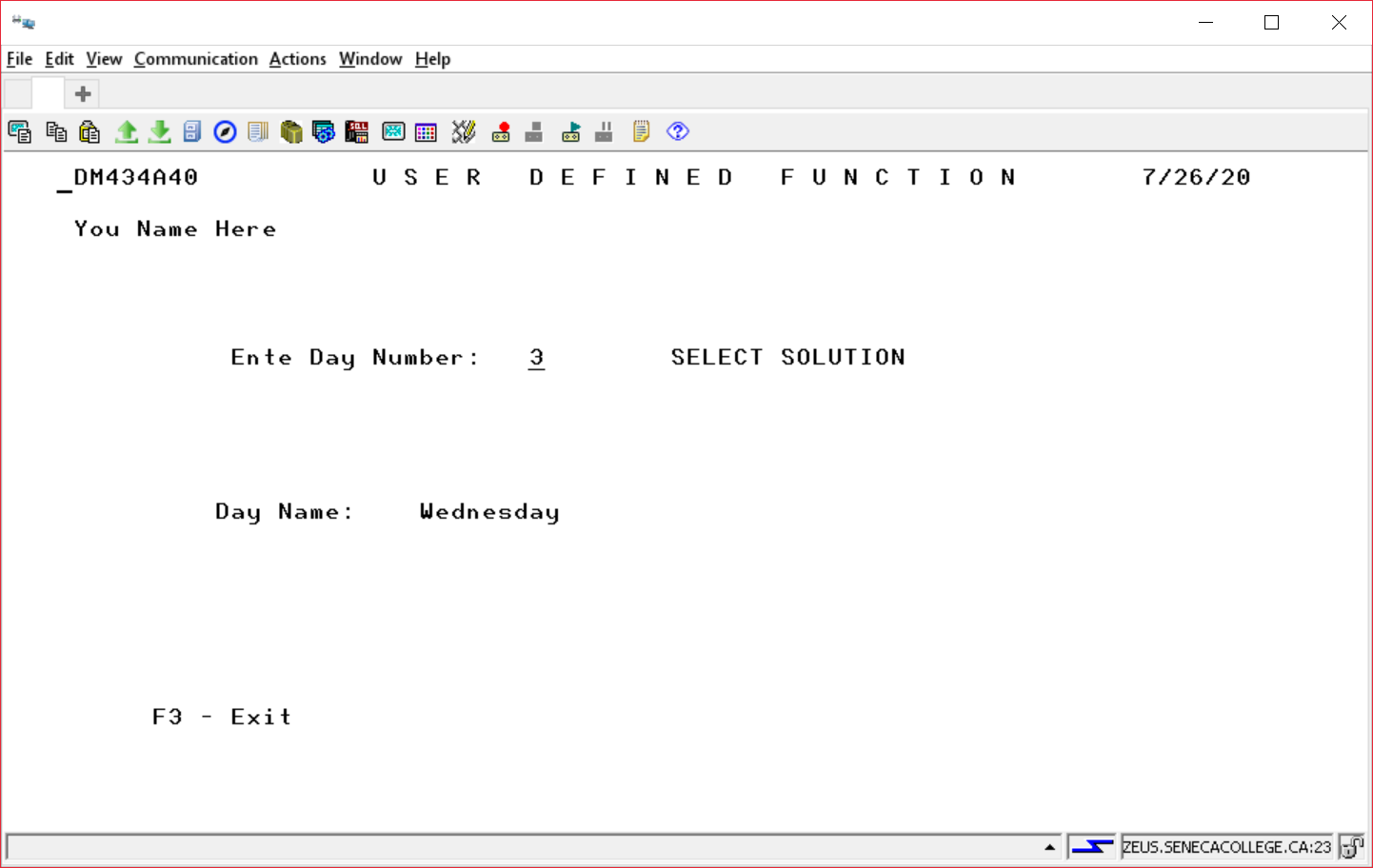


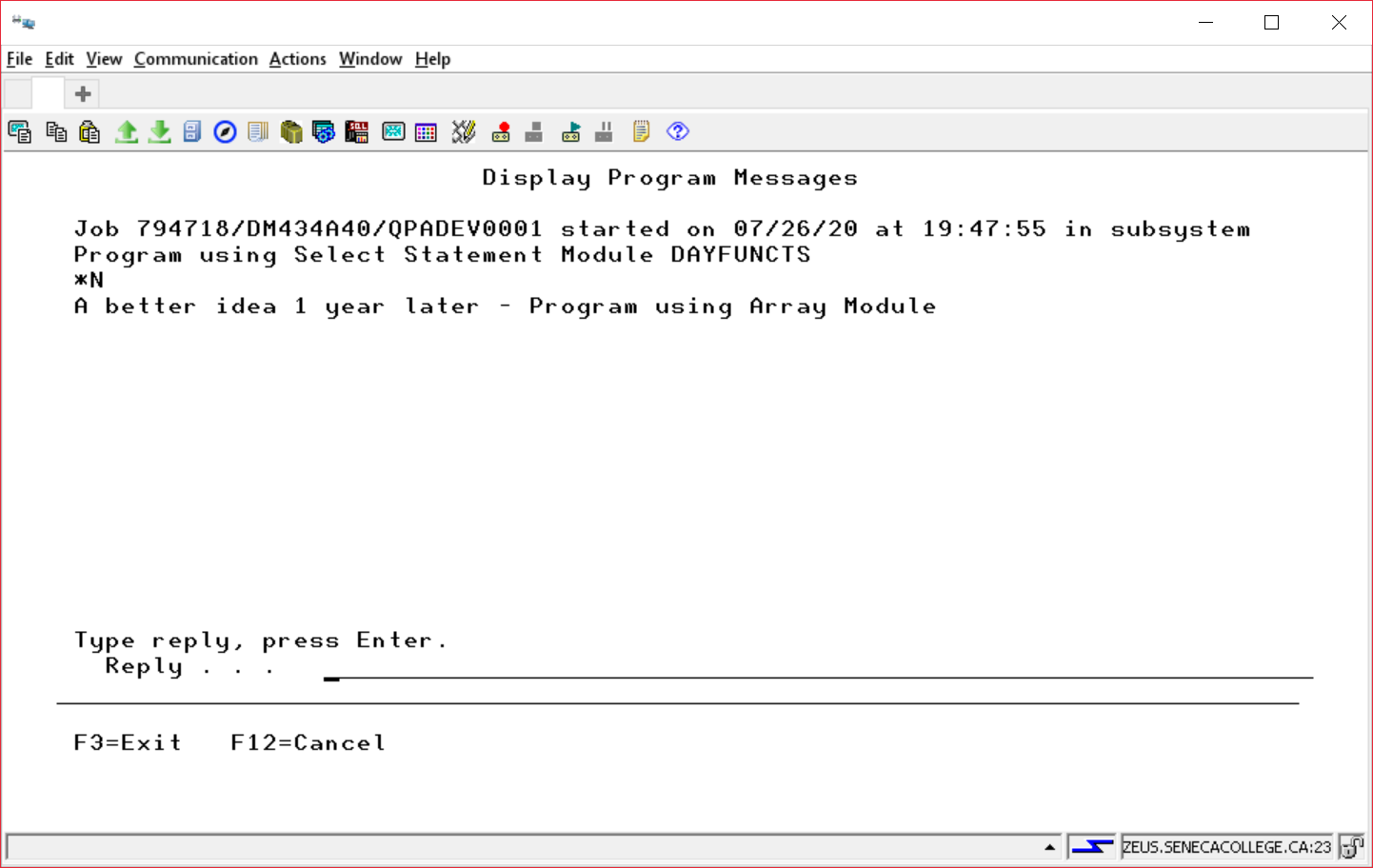


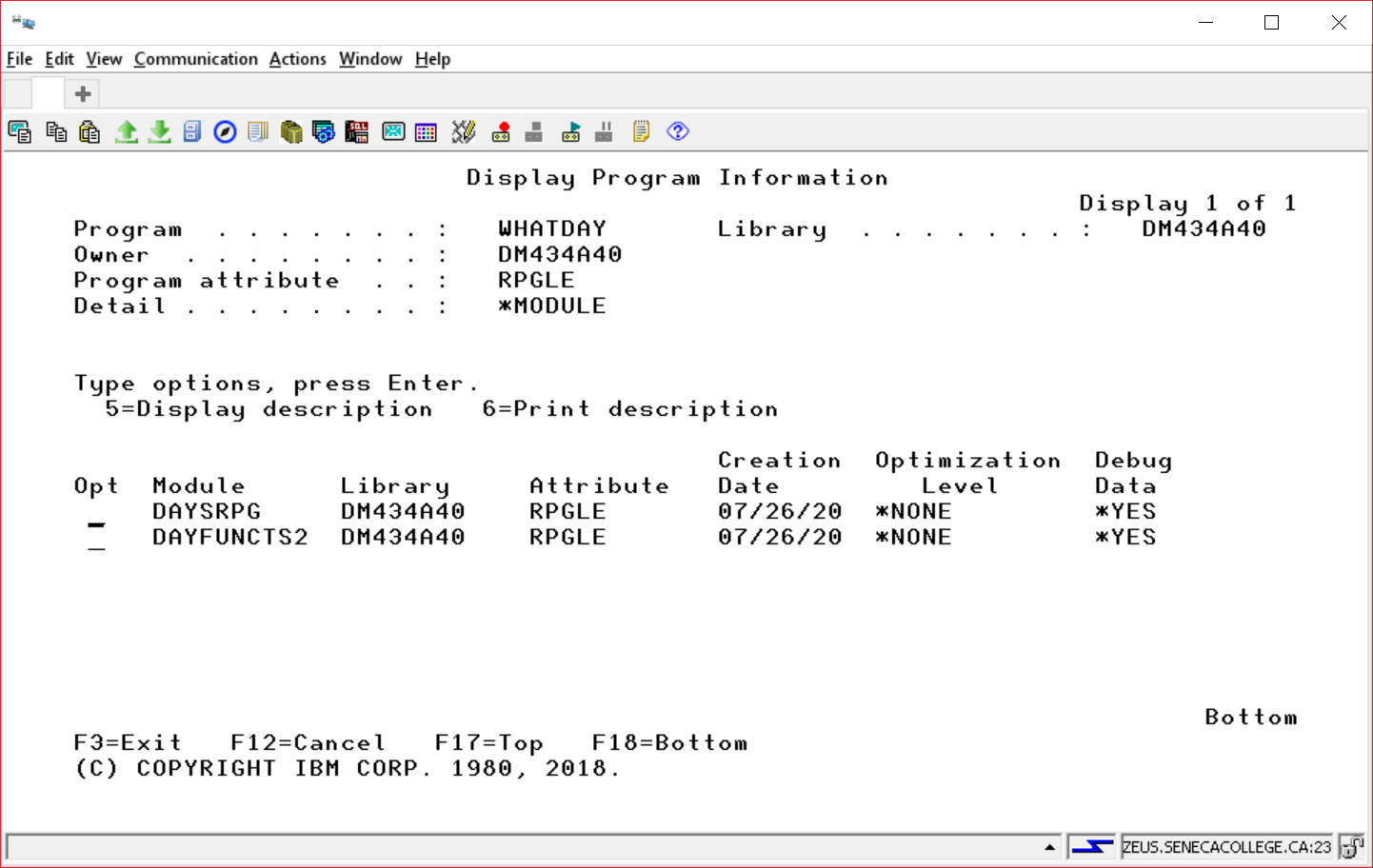
This wasn’t shown in class. Look closely at the screen shot to see what command may work.



3 was entered and then the enter key was pressed to get the result.







You would show your module code for DAYFUNCTS2 (just like the code showed for DAYFUNCTS)

instead of using SNDUSRMSG to say “Show your array module code here”.

