CALIFORNIA STATE UNIVERSITY, LONG BEACH

**IS 482 – Enterprise Systems**

## Spring 2023 Term – Session 01 (#9877) – Individual Assignment 9

**Notes:**

* Submit solution via Canvas
* There will be a quiz based on this material
* Reference Canvas for connection information. Use the following:

**Product: SAP ABAP - S/4 Development**

System: **MER**

Client: **105**

Server: **MERIDA.COB.CSUCHICO.EDU**

System Number: **38**

Web URL: <https://merida.cob.csuchico.edu:8038/sap/bc/gui/sap/its/webgui/>?

Student Accounts

Username: CSULB-### (### represents the number assigned to you, e.g. 033)

Password: global7 (This is the initial password, you will need to change it at login)

**Tutorial 4 – ABAP4 Subroutines**

Turn in the following for each exercise:

* 1. Code listing (Text. This can be copy/pasted to test the code)
  2. Results screen shot

**Exercises**

For the exercises below, use a subroutine to break up the code (modularize it).

Review the requirements of each subroutine:

The inputs

The returned value

The subroutine is to return value requested only. It should not perform any other tasks.

In addition to modularizing the code, this will allow for code reuse.

1. Recall ZZ\_###\_STUDENTLEVEL that you created in a previous assignment:

User inputs how many units they have earned.

The program outputs their student level as follows:  
0 – 30 units earned – Freshman

31 – 60 units earned – Sophomore

61 – 90 unites earned – Junior

> 90 – Senior

If a negative number is entered, the program writes:  
Invalid input. Units earned must be a positive number.

Create a revised version that uses a subroutine: ZZ\_###\_STUDENTLEVEL\_SUB

Your main program will call a subroutine that performs the following:

Input: Number of Units  
Output: Student Level.

NOTE:

Your subroutine is to return the student level only.

It does not perform any other tasks: data validation, output to user, etc.

Your main program is to treat the subroutine as a “black box”.

You provide the requested input (number of units) and it returns the student level.

1. a.

Create a program to print the first fifty prime numbers: ZZ\_###\_PRIMENUMBERS.

Your main program is to loop until you get 50 prime numbers. The main program will call a subroutine that performs the following:

Input: A number

Output: A single character

‘Y’ - Yes, the number is prime

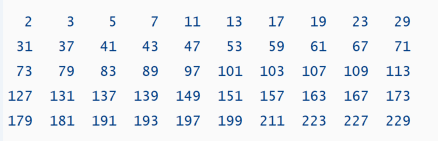
‘N’ – No, the number is not prime.

Again, your subroutine is a “black box”.

Input is a number, and it returns “Y” or “N”.

It does not do anything else: validations, output to the user, etc.

Output 10 prime numbers per line. See below output:



b. Enhance the output to emulate boxes around each number (use the pipe and dash to emulate lines: | and -). Enhanced output:

