**CSCI 360-1 Assignment 11 – Tables and DSECTs Fall 2018**

**(Final Exam Project)**

**200 exam/quiz points**

**Overview**

This assignment adds functionality to your Assignment 10 program and constitutes your final exam.

To start, copy your ASSIGN10 PDSE member and name the new member ASSIGN11. Name the main, or driver, program's CSECT (and the END statement at the very end) PAYROLL as you did for Assignment 10. Leave the names of the subprograms the same as before.

The input data set is:

DSN=KC02322.CSCI360.DATAFA18(DATA9),DISP=SHR

Each record has the same format as before.

This program will produce the same report as that created in your Assignment 10 program but the processing of the input data set of employee payroll information will be different. In general, you will first read the input data set from beginning to end, one record at a time, and store each employee's payroll information into an entry in a table you have defined in the main routine's storage.

You will then loop through the table and process each employee's payroll information as stored in the table, producing the same report you created in Assignment 10. The body of the second loop will be very much like what you did in the read loop in Assignment 10.

**Programming Notes**

Declare a table in the storage of PAYROLL named EMPLTBL that can store up to 100 entries of 53 bytes each. Initialize each and every byte of the table with an "at" sign, i.e., @.



Because of the size of the table, you will need to establish a second base register for your program. Copy the following four lines into the PAYROLL program immediately following standard entry linkage.

LR 11,12

LA 11,4095(,11)

LA 11,1(,11)

USING PAYROLL+4096,11

This means that you will be unable to use register 11. You may only use registers 2 through 10 except in special instances.

Create a new additional subprogram named BUILDTBL that will be passed two parameters when called: 1) the address of EMPLTBL and 2) the address of the 3-byte packed decimal field you use to keep track of the number of employees.

The read loop for the input file will now be moved into the BUILDTBL subprogram. The first 53 bytes of each input record will need to be moved into the table, one entry after the other. For this you can use a single MVC.



When all of the records from the input file have been read and copied into the table, you will return to PAYROLL and the number of employees 3-byte packed decimal variable should hold an integer value representing not only the number of employees but also the number of entries in the 100-entry table that are "filled" with information.

Declare a dummy section, or DSECT, to give names to the variables stored for each employee in each entry of the table. It will be easiest to base your DSECT on the following "description" of the first 53 bytes of each of the input records that was given to you in Assignment 8:

INPUTREC DS 0H

IEMPID DS ZL8

IHRPAY DS ZL5

IHOURS DS ZL5

IDEDUCT DS ZL5

IBONUS DS ZL5

IEMPNME DS CL25

As you process the table entries in the second loop, you will process each employee's payroll information like you did in Assignment 10. Note that there will be NO packing of data in the BUILDTBL routine. In fact, the read loop in BUILDTBL will only have four instructions in the body as follows in pseudocode:

read first record

\*

top-of-loop is it end-of-file? if so, branch to end-of-loop

\*

1) add 1 to employee counter

2) move employee information into table entry

3) increment table pointer register

4) read next record

\*

ranch to top-of-loop

\*

end-of-loop return to caller

Your output report should look identical to that produced by your Assignment 10 program.

Document your program completely according to Chapter 2 of the CSCI 360 Course Notes and submit your single .txt file on Blackboard as before.

**Note that absolutely NO late assignments 11 will be accepted! NO EXCEPTIONS! DON'T ASK!**