**CSCI 360-1/PE1 Assignment 6 – Multiplication and Division Spring 2022**

**125 points**

Write an Assembler program to calculate payroll for a business. Each employee has an hourly pay rate, a number of hours worked in the pay period, a deduction amount for any lunches eaten in the company cafeteria during the pay period, and, if earned, a bonus amount. Of course, each employee has a name and employee identification number.

A .txt document named 360 Assign 6 Starter Program Sp22.txt is provided for you to begin this assignment. Copy it into IDz and save as a new member of your ASSIGNS PDSE named ASSIGN6 to begin this assignment. Be sure to enter your KC-ID in the upper left hand corner and put your first initial, a period, and your last name in between the tick marks in the first line, the JOB card.

The first thing the executable code provided does is load register 10 with the federal tax rate percentage and load register 11 with the state tax rate percentage. These two registers will be used in your arithmetic so do not accidentally overwrite them or they will be irretrievable.

**Input**

Each of the input records contains, from left to right across the 80-byte record, the following data about a single employee: employee name, employee id number, hourly pay rate, number of hours worked in the pay period, deduction amount, and bonus amount, with the three dollar amounts rounded to the whole dollar. Remember to use MVC to move character data (even thought it is numeric, you may consider the employee id number character data – people do not do arithmetic with ID numbers…usually).

**Calculations**

Your program must calculate each employee's gross pay amount using the following formula:

gross pay amount = hourly pay rate \* number of hours worked –

deduction amount + bonus amount

And then calculate each employee's net pay amount using the following formula:

net pay amount = gross pay amount -

(gross pay amount \* current federal withholding percentage / 100) -

(gross pay amount \* current state withholding percentage / 100)

Your program also must count the number of employees processed and keep a running total of each of the gross pay amount, federal withholding amount, state withholding amount and net pay amount.

Note: When doing division, disregard the remainder and use only the quotient of the division. Working with decimal points will be included in a future assignment.

Be VERY smart about your register usage. For example, after you get the gross pay calculated and added to the total gross pay amount register, you can reuse the registers that were used to hold hourly pay rate, number of hours worked, deduction amount and bonus amount (of course, you also need to have XDECO'd those values into the print line and have added the deduction and bonus amount registers to their total registers too).

(continued)

**Output**

For each record, print out (using XPRNT) the employee's id number, employee name, hourly pay rate, number of hours worked, gross pay amount, federal withholding amount, state withholding amount and, finally, net pay amount, that you calculated across one output line, double spaced.

At the end of the report, and on separate lines and on the left, just under the employee's ID, print out the number of employees processed, total gross pay, total federal withholding amount, total state withholding amount and total net pay amount. Be sure that you double space all of the output lines.

For guidance, see the exact output document named:

**Note that employee ID begins in column 26 and the first numeric value for calculating pay begins in column 32 of each input record.**

360 Assign 6 Exact Output Sp22.txt

**Important Notes**

Here is the data in the PDSE member

named DATA6:

WOLFGANG AMADEUS MOZART **1**2345 **1**5 80 25 500

RICHARD STRAUSS 23456 19 80 00 010

AMY BEACH 22132 10 80 31 200

DAME ETHEL SMYTHE 65465 22 80 15 900

PETER ILYICH TCHAIKOVSKY 44560 23 28 34 070

ANTON BRUCKNER 99870 22 80 21 000

LUDWIG VAN BEETHOVEN 13345 15 80 25 500

JOHANNES BRAHMS 24456 19 80 00 010

BELA BARTOK 22532 10 80 31 200

MAX REGER 11465 22 80 15 900

SAMUEL BARBER 13360 23 28 34 070

GIUSEPPE VERDI 99873 22 80 21 000

JOHANN SEBASTIAN BACH 12342 15 80 25 500

JOSEPH HAYDN 23452 19 80 00 010

GEORG FRIEDRICH HANDEL 22131 10 80 31 200

EDWARD ELGAR 65411 22 80 15 900

NIKOLAI RIMSKY-KORSAKOV 44378 23 28 34 070

CLAUDE DEBUSSY 99855 22 80 21 000

ANTONIN DVORAK 13346 15 80 25 500

THOMAS TALLIS 24457 19 80 00 010

RALPH VAUGHAN WILLIAMS 22538 10 80 31 200

RICHARD WAGNER 11477 22 80 15 900

FREDERIC CHOPIN 17777 23 28 34 070

JOAN TOWER 99211 22 80 21 000

(Note that you can use a label followed by a plus sign and an integer value in any instruction that requires a D(X,B) or D(B) type address to represent a displacement from the address of the label. For example: RECORD+25 refers to the 26th byte of the RECORD and RECORD+31 refers to the 32nd byte of the RECORD.)

You may use registers 2 through 11, inclusive. It is suggested that you use register 6 as your employee (record) counter and 7, 8 and 9 as accumulator (totals) registers. These four must be zeroed out prior to starting the read loop. No other registers need be cleared before processing begins.

After you MVC both the employee ID and name to the print line, here is a bit of a start of the calculations to give you an idea of how to proceed. You will need to use registers 2 through 5 very judiciously.

XDECI 2,RECORD+31 GET PAY RATE

XDECO 2,OPAYRATE DISPLAY PAY RATE

XDECI 3,0(1) GET HOURS

XDECO 3,OHOURS DISPLAY HOURS

\*

MR 2,2 PAY RATE \* HOURS (PRODUCT IN R3)

\*

XDECI 2,0(1) GET DEDUCTIONS

SR 3,2 SUBTRACT FROM PAY

XDECI 2,0(1) GET BONUS

AR 3,2 ADD TO PAY

\*

XDECO 3,OGROSS DISPLAY GROSS PAY AMT

\*

From this point on, you will need to calculate federal withholding, display it and add it to the totals register for federal withholding, then state withholding, display it and add it to the totals register for state withholding. After this, subtract those two amounts from the gross pay amount and display it. You ARE allowed to use a single fullword of storage to perhaps save the gross pay amount but you should be able to do everything without it.

If you find that you run out of registers to do the accumulating, you can always declare the totals fields as fullwords in storage. Be sure you declare them as DC F'0' to initialize them. You can use them to add to a register that holds the value you have calculated and then store that register back into the totals field you just added.

**Other**

Fully document your program as instructed in the **CSCI 360 Coding and Documentation Guidelines** soon to be found in Blackboard's Course Documents.

Be sure you are using the structured read loop as taught to you in class.

Be sure that your ENTIRE output is included in the .txt file BEFORE you submit it for grading. If any or all of it is missing, you will earn a 0 on the assignment.

Submit the .txt output from Marist on Blackboard as before.