CSCI 360 **Assignment 6 – Multiplication and Division** **Summer 2020**
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.

To begin this assignment, you can use TSO/ISPF option 3.3 to copy a previous assignment member in your `ASSIGNS` Partitioned Data Set Extended (PDSE) and name the new copied member `PAYROLL1`. If you do this, be sure to change `ASSIGNn` to `PAYROLL1` in the source code wherever necessary.

Input

The first input record has a single integer representing the current tax withholding percentage. Read that record and get the percentage into a register to use in arithmetic in the body of the loop.

Each of the rest of the input record 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 withholding percentage / 100)
```

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

Finally, after the loop ends, your program must calculate the average gross pay and net pay amount for the employees listed. Use the following formulas:

```
average gross pay amount = total gross pay amount / employee count

average net pay amount = total net pay amount / employee count
```

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).

Output

For each record, print out (using `XPRNT`) the employee's name, employee id number, hourly pay rate, number of hours worked, deduction amount, bonus amount, gross pay amount, 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, the total deduction amount, total bonus amount, total gross pay, total withholding amount and total net pay amount. And finally, on a fifth line, print out the average gross pay amount and average net pay amount. Be sure that you double space all of the output lines.

For guidance, see the exact output document named:

`360 Assign 5 Exact Output Su20.txt`

Important Notes

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

Here is the data in the PDSE member:

```
18
WOLFGANG AMADEUS MOZART  12345 15 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: `BUFFER+25` refers to the 26th byte of the `BUFFER` and `BUFFER+31` refers to the 32nd byte of the `BUFFER`.)

The data above can be accessed by copying or typing in the following into your `PAYROLL1` PDSE member replacing what is there from your previous assignment's PDSE member:

```
         END   PAYROLL1   ← provided as a reference to help you place the following JCL.
/*
//*
//FT05F001 DD DSN=KC02322.CSCI360.DATASU20(DATA5),DISP=SHR
//*
//FT06F001 DD SYSOUT=*
//*
//SYSPRINT DD SYSOUT=*
//
```

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.