CSCI 360 **Assignment 8: Introduction to Packed Decimal**
75 points

This program is a rather simple introduction to packed decimal and packed decimal arithmetic.

First, copy the following Assembler program into a new member of your `ASSIGNS` PDSE named `ASSIGN8`.

Be sure the program is surrounded by the correct `JCL` as before:

```
         PRINT NOGEN
*****************************************************************
*                                                               *
*  CSCI 360               ASSIGNMENT 8          current semester  *
*                                                               *
*                                                               *
*                                                               *
*****************************************************************
*
PAYROLL2 CSECT
*
*  STANDARD ENTRY LINKAGE WITH R12 AS BASE REGISTER
*
         STM   14,12,12(13)  SAVE REGS IN CALLER'S SAVE AREA
         LR    12,15         COPY CSECT ADDR INTO R12
         USING PAYROLL2,12   ESTABLISH R12 AS THE BASE REG
         LA    14,SAVEREGS   R14 POINTS TO THIS CSECT'S SAVE AREA
         ST    14,8(,13)     STORE ADDR OF THIS CSECT'S SAVE AREA
         ST    13,4(,14)     STORE ADDR OF CALLER'S SAVE AREA
         LR    13,14         POINT R13 AT THIS CSECT'S SAVE AREA
*
*
*  Here is where you will write your program.
*
*
*  STANDARD EXIT LINKAGE WITH RC OF 0
*
         SR    15,15        R15 = RETURN CODE OF 0
         L     13,4(,13)    POINT R13 TO CALLER'S SAVE AREA
         L     14,12(,13)   RESTORE REGISTER 14
         LM    0,12,20(13)  RESTORE R0 THRU R12
*
         BR    14           RETURN TO CALLER
*
         LTORG
*
SAVEREGS DS    18F          PROGRAM'S REGISTER SAVE AREA
```

```
*
*  Here is where your storage will be defined.
*
         END   PAYROLL2
```

You will write your source code in between the lines standard entry and exit linkage code provided to you. Of course, you can add variables and other storage declarations immediately following the `LTORG` provided but DO NOT place them above the 19-fullword caller's register storage area.

Secondly, change the input data member to:

```
//FT05F001 DD DSN=KC02322.CSCI360.ASNDATA(DATA8),DISP=SHR
```

To assist you, here is the data in the `DATA8` member:

```
WOLFGANG AMADEUS MOZART   123450158002550
RICHARD STRAUSS           234560198007010
AMY BEACH                 221320108003120
DAME ETHEL SMYTHE         654650228001590
PETER ILYICH TCHAIKOVSKY  445600232803407
ANTON BRUCKNER            998700228002100
LUDWIG VAN BEETHOVEN      133450158002550
JOHANNES BRAHMS           244560198008001
BELA BARTOK               225320108003120
MAX REGER                 114650228001590
SAMUEL BARBER             133600232803407
GIUSEPPE VERDI            998730228002100
JOHANN SEBASTIAN BACH     123420158002550
JOSEPH HAYDN              234520198002001
GEORG FRIEDRICH HANDEL    221310108003120
NIKOLAI RIMSKY-KORSAKOV   443780232803407
EDWARD ELGAR              654110228001590
CLAUDE DEBUSSY            998550228002100
ANTONIN DVORAK            133460158002550
THOMAS TALLIS             244570198006001
RALPH VAUGHAN WILLIAMS    225380108003120
RICHARD WAGNER            114770228001590
FREDERIC CHOPIN           177770232803407
JOAN TOWER                992110228002100
```

Each of the above input records has the following layout:

Data Field	Data Type	Columns	Max. Value
Employee Name	Alphanumeric	1-25	
Unused		26	
Employee ID	Alphanumeric	27-31	
Hourly Pay Rate	Zoned Decimal	32-36	$999.99 (two implied decimal places)
Hours Worked	Zoned Decimal	37-41	999.99 (two implied decimal places)
Unused		42-80	

As before, read the employee records one at a time using a standard read loop and printing a detail line for each employee so that your output looks like that provided to you in the Assignment 8 folder on Blackboard.

For each employee, move their name to the print line and move their ID to the print line. Make sure your read loop works at this point before moving on.

Next, using only packed decimal instructions, `PACK` and `ED` the Hourly Pay Rate into the print line showing the decimal point. Do the same for the Hours Worked.

Finally, calculate the gross pay amount using `MP`, round it to two decimal places and `ED` it into the print line showing the decimal point.

The only packed decimal instructions you will need are: `PACK`, `ED`, `ZAP`, `MP` and `SRP`.

You will no longer need to use `XDECI` or `XDECO`. DO NOT use them!

Name your detail line you print for each employee `DETAIL` and place the following two lines at the top of your read loop:

```
MVI   DETAIL+1,C' '
MVC   DETAIL+2(131),DETAIL+1
```

These two lines set all but the very first byte – the carriage control character – of your 133-byte detail line to spaces to prepare for the next employee information.

The JCL statements ***immediately*** following the `END` statement of the program should look like this:

```
/*
//*
//FT05F001 DD DSN=KC02322.CSCI360.ASNDATA(DATA8),DISP=SHR
//*
//FT06F001 DD SYSOUT=*
//
```

Check your output against the provided file `360 Assign 8 Exact Output.txt`.

Fully document your program as instructed in the **CSCI 360 Coding and Documentation Guidelines** found in Blackboard's Course Documents. Be sure every line of code has adequate line documentation.

Submit your `ASSIGN8.txt` file on Blackboard as before.