Assignment 3

100 points

Overview

For this assignment, write a program that will produce a sales report for a company called Widgets-R-Us. They are offering discounts and free shipping for sufficiently large orders. For one product, there is a promotional discount.

Input

The input to the program will be a file with an unknown number of records. Each record represents a single sale and has the following format:

columns	description
1 - 6	Product ID
7 - 8	spaces
9 - 15	Order ID
16 - 17	spaces
18 - 20	Number Ordered
21 - 22	spaces
23 - 26	Price Each (in cents)
27 - 80	spaces

Use the following JCL statement to specify the input file:

```
//FT05F001  DD  DSN=KC02314.SPRING19.CSCI360.HW3DATA,DISP=SHR
```

Processing Requirements

For this assignment, write a read loop that will process the input one record at a time.

For each input record, print a detail line containing the input values along with the Discount, Shipping and Total Cost (in cents).

Use the following formulas for your calculations:

```
PreTotal = Price Each * Number Ordered

If (PreTotal >= 15000)
  Then Discount = 500
  Else Discount = 0
End If

If (Product ID == 566824)
  Then Add 1000 to Discount
End If

If (PreTotal >= 5000)
  Then Shipping = 0
  Else Shipping = 600
End If

Total Cost = PreTotal + Shipping - Discount
```

As you process records, count them and accumulate the Grand Total = sum of Total Cost for all sales. Also count sales for which Shipping = 0.

After processing all of the input records, print summary lines which tell us the number of sales, the number of sales with Shipping = 0, the Grand Total and the average Total Cost per Sale.

The average Total Cost should be rounded to the nearest cent. One possible way to do this is, after doing the division, calculate 2 times the remainder and compare that product with the divisor. If 2 * the remainder is greater than or equal to the divisor, then the quotient should be rounded up up by 1.

Other Requirements

1. Use `TITLE 'Your Name, CSCI 360, Program 3'`.

2. Use `M, D, MR, DR, LA, XREAD, XDECI, XDECO, XPRNT, L, ST, C, A, S, AR, SR, BC` and `BCR`, as appropriate.

3. Use at least one literal in the program.

4. You should use line documentation, documentation boxes, and you are welcome to use `EJECT`, and `SPACE`. You will be using various registers, and you do need to make a list of how you are using them.

5. Write this program incrementally. This will make it much easier to debug. Initially write the program so that it reads and processes one record. Once you know it does that correctly, implement the loop. Once the loop is working correctly, add the code to count the sales, count those with free Shipping, etc.

6. You may find that you run short of registers if your try to use them for all your variables. You can use fullwords in memory for some purposes such as counters. When you want to increment the counter, load it into a register, add 1 and store it again. That register is not then tied up and can be used for other purposes.

7. The JCL for the program should now look like:

   ```
   //your_KC_id_plus_a_letter JOB ,'your name',MSGCLASS=H
   //STEP1 EXEC PGM=ASSIST
   //STEPLIB DD DSN=KC02293.ASSIST.LOADLIB,DISP=SHR
   //SYSPRINT DD SYSOUT=*
   //SYSIN DD *
   ********************************************
   *
   *  The documentaton goes here.
   *
   ********************************************
   /*
   //FT05F001 DD DSN=KC02314.SPRING19.CSCI360.HW3DATA,DISP=SHR
   //
   ```

8. When you are debugging logic errors, don't be reluctant to use XDUMP instructions to display registers and/or memory. Once you get the program running, remove the XDUMPs from the run that is turned in for grading.

   Feel free to test the program with your own data. To do so, temporarily replace the JCL line that starts with "//FT05F001" by

   ```
        //FT05F001 DD *
        your data goes here
        /*
   ```

   Make sure that the run you submit for grading uses the file specified above.

Other Notes

The page header for the report should start at the top of a new page. The column headers should be double spaced from the page header. The sales information lines should be double spaced. The summary lines should be triple spaced from the last sales record. Double-space between summary lines.

Notice that we are printing amounts of money simply as numbers of cents. Later in the course, we will have ways to format numbers more precisely.

You may need to put a label on a line now and then when you are not yet sure what instruction should be on that line. In that case, you can just use:

```
MYLABEL  DS   0H
```

The "DS 0H" part allocates zero halfwords on a halfword boundary. Instruction are always on a halfword boundary anyway, so this does no harm and simply associates MYLABEL with an address. The next line will be at the same LOC value.