CS 243 Program 1 (100 pts)

Due date: Time printed on Job Listing must be before 8:30am Thu Oct 2
If not handed in on time, you go into the late schedule (see syllabus)
Final Due date: Time printed on Job Listing must be before 8:30am Mon Oct 6
Listings must be in my office before 3:00pm the next class day
You may work alone or in pairs on programming assignments all semester (your choice, see syllabus).

Write a COBOL Program to print a loan repayment schedule.

Use a starting balance of $5,000, an annual interest rate of 8%, and a term of 3 years. These three values are to be specified in the VALUE clauses of the data items BALANCE, ANNUAL-RATE and YEARS, respectively. Monthly payments are to be made. Use data names PAYMENT, MONTHS, RATE and INTEREST in addition to the names already described. Write the program so that in the Procedure Division you always use data names for values instead of numeric literals such as 5000, 0.08, 3, etc.

Declare RATE to be RATE PIC 9V9(15).
Declare BALANCE to be BALANCE PIC 9(6)V99.

Use the DISPLAY statement for output; your output must look as follows. Suppress leading zeros for the monthly payment. Use a fixed $ for the interest column and a floating $ for the balance column. Use a month column and suppress leading zeros. Use a floating $ for the Accumulated Interest.

MONTHLY PAYMENT
  156.69

  MONTH     INTEREST      BALANCE
  1       $ 33.33    $5000.00
  2       $ 32.51    $4876.64
         etc.
  35      $  2.07    $155.32
  36      $  1.04    $ 0.00

ACCUMULATED INTEREST
  $dd,ddd.dd

If you were to solve this problem by hand, with the aid of a calculator, you would perform the following steps (algorithm).

1. compute monthly interest: \[ 5,000 \times r = 33.33 \] (rounded)
2. determine amount used to reduce the loan: \[ 156.69 - 33.33 = 123.36 \]
3. determine new balance: \[ 5,000 - 123.36 = 4876.64 \]
repeat
1. compute monthly interest: \[ 4876.64 \times r = 32.51 \] (rounded)
2. determine amount used to reduce the loan: \[ 156.69 - 32.51 = 124.18 \]
3. determine new balance: \[ 4876.64 - 124.18 = 4752.46 \]
repeat until done
The monthly payment can be computed as follows.
Let \( b = \text{balance}, \ a = \text{annual interest rate}, \ y = \text{number of years}, \ m = \text{number of months} \) and \( r = \text{monthly interest rate} \). Then, \( m = 12y, \ r = a/12 \) and the monthly payment, \( p \), would be computed by the algebraic formula

\[
p = \frac{b \cdot r \cdot (1 + r)^m}{(1 + r)^m - 1}
\]

But, in order to get the correct answer, you need to do the following.
(1) Calculate the monthly payment but do not use ROUNDED;
(2) add 0.01 to the calculated monthly payment,

\text{\textit{i.e.}}

\[
\text{COMPUTE PAYMENT =} \\
\text{ADD .01 TO PAYMENT}
\]

Then, use a loop to print all lines except the last line. Do a separate calculation to get the last line as a special case.

The idea is as follows:
The monthly payment in this example is something like 156.682. Using 156.68 would leave a balance at the end of the period because if you need to pay 156.682 to get down to zero, then paying only 156.68 will not get the balance down to zero. So, we use a monthly payment exactly \$.01 larger, 156.69. This means that we will owe less than 156.69 the last month in order to bring the balance down exactly to zero. This strategy will work most of the time because it is unlikely that the monthly payment would be calculated exactly to two decimal places mathematically.

\text{MEMBER Name for Cobol Source Program: PGM1} \\
\text{MEMBER Name for Jcl File} : \ JCL0 \text{ or } JCL1 \\
(either modify JCL0 or create a new JCL1, your choice)

\text{REMEMBER}
Never leave listings in the lab. See the syllabus.