COMP2710: Homework 2

Description:

You have just purchased a stereo system that cost $1000 on the following credit plan: no down payment, an interest rate of 18% per year (and hence 1.5% per month), and monthly payments of $50. The monthly payment of $50 is used to pay the interest, and whatever is left is used to pay part of the remaining debt. Hence, the first month you pay 1.5% of $1000 in interest. That is $15 in interest. The remaining $35 is deducted from your debt, which leaves you with a debt of $965.00. The next month you pay interest of 1.5% of $965.00, which is $14.48. Hence, you can deduct $35.52 (which is $50–$14.48) from the amount you owe.

Write a program that will tell you how many months it will take you to pay off this loan in particular and any loan in general. Your program also needs to calculate the total amount of interest paid over the life of any loan. Use a loop to calculate the amount of interest and the size of the debt after each month. Your program must output the monthly amount of interest paid and remaining debt. Use a variable to count the number of loop iterations and hence the number of months until the debt is zero. You may want to use other variables as well. The last payment may be less than $50 if the debt is small, but do not forget the interest. If you owe $50, then your monthly payment of $50 will not pay off your debt, although it will come close. One month's interest on $50 is only 75 cents.

Here is a sample dialog (where the user input is depicted as **Bold**, but you do not need to display user input in bold.):

```
Loan Amount: 1000
Interest Rate (% per year): 18
Monthly Payments: 50
```

```
*****************************************************
        Amortization Table
*****************************************************
Month   Balance Payment Rate    Interest Principal
0       $1000.00 N/A    N/A     N/A      N/A
1       $965.00 $50.00  1.5     $15.00   $35.00
2       $929.48 $50.00  1.5     $14.48   $35.53
3       $893.42 $50.00  1.5     $13.94   $36.06
4       $856.82 $50.00  1.5     $13.40   $36.60
5       $819.67 $50.00  1.5     $12.85   $37.15
6       $781.97 $50.00  1.5     $12.30   $37.70
7       $743.70 $50.00  1.5     $11.73   $38.27
8       $704.85 $50.00  1.5     $11.16   $38.84
9       $665.42 $50.00  1.5     $10.57   $39.43
10      $625.40 $50.00  1.5     $9.98    $40.02
11      $584.79 $50.00  1.5     $9.38    $40.62
12      $543.56 $50.00  1.5     $8.77    $41.23
13      $501.71 $50.00  1.5     $8.15    $41.85
14      $459.24 $50.00  1.5     $7.53    $42.47
15      $416.13 $50.00  1.5     $6.89    $43.11
16      $372.37 $50.00  1.5     $6.24    $43.76
17      $327.95 $50.00  1.5     $5.59    $44.41
18      $282.87 $50.00  1.5     $4.92    $45.08
19      $237.11 $50.00  1.5     $4.24    $45.76
20      $190.67 $50.00  1.5     $3.56    $46.44
21      $143.53 $50.00  1.5     $2.86    $47.14
22      $95.68  $50.00  1.5     $2.15    $47.85
23      $47.12  $50.00  1.5     $1.44    $48.56
24      $0.00   $47.83  1.5     $0.71    $47.12
*****************************************************

It takes 24 months to pay off the loan.
Total interest paid is: $197.83
```

Your program's output should match the style of the sample output.

In what follows, you find a second case used to test the correctness of your program.

```
Loan Amount: 2000
Interest Rate (% per year): 12
Monthly Payments: 80
```

```
*****************************************************
        Amortization Table
*****************************************************
Month   Balance         Payment Rate    Interest Principal
0       $2,000.00       N/A     N/A     N/A     N/A
1       $1,940.00       $80.00  1       $20.00  $60.00
2       $1,879.40       $80.00  1       $19.40  $60.60
3       $1,818.19       $80.00  1       $18.79  $61.21
4       $1,756.38       $80.00  1       $18.18  $61.82
5       $1,693.94       $80.00  1       $17.56  $62.44
6       $1,630.88       $80.00  1       $16.94  $63.06
7       $1,567.19       $80.00  1       $16.31  $63.69
8       $1,502.86       $80.00  1       $15.67  $64.33
9       $1,437.89       $80.00  1       $15.03  $64.97
10      $1,372.27       $80.00  1       $14.38  $65.62
11      $1,305.99       $80.00  1       $13.72  $66.28
12      $1,239.05       $80.00  1       $13.06  $66.94
13      $1,171.44       $80.00  1       $12.39  $67.61
14      $1,103.15       $80.00  1       $11.71  $68.29
15      $1,034.19       $80.00  1       $11.03  $68.97
16      $964.53         $80.00  1       $10.34  $69.66
17      $894.17         $80.00  1       $9.65   $70.35
18      $823.12         $80.00  1       $8.94   $71.06
19      $751.35         $80.00  1       $8.23   $71.77
20      $678.86         $80.00  1       $7.51   $72.49
21      $605.65         $80.00  1       $6.79   $73.21
22      $531.70         $80.00  1       $6.06   $73.94
23      $457.02         $80.00  1       $5.32   $74.68
24      $381.59         $80.00  1       $4.57   $75.43
25      $305.41         $80.00  1       $3.82   $76.18
26      $228.46         $80.00  1       $3.05   $76.95
27      $150.75         $80.00  1       $2.28   $77.72
28      $72.25          $80.00  1       $1.51   $78.49
29      $0.00           $72.98  1       $0.72   $72.25
*****************************************************

It takes 29 months to pay off the loan.
Total interest paid is: $312.98
```

Special Cases:

1. Please think about how to deal with the last payment, which is smaller than regular payment. For example, in the above table, the regular payments are $80.00 whereas the last payment is only 72.98.
2. Your program needs to ensure that regular payments are larger than any monthly interest. For example, in the above amortization table, your program must test if the regular monthly payment (i.e., $80.00) is larger than the monthly interest (e.g., $20.00 in the first month).
3. If you do not address the above issue, your program may not terminate in some special cases. See the example below:

Loan Amount: **2000**
Interest Rate (% per year): **49.2**
Monthly Payments: **80**

****
Amortization Table
****

Month	Balance	Payment	Rate	Interest	Principal
0	$2,000.00	N/A	N/A	N/A	
1	$2,002.00	$80.00	4.1	$82.00	-$2.00
2	$2,004.08	$80.00	4.1	$82.08	-$2.08
3	$2,006.25	$80.00	4.1	$82.17	-$2.17
4	$2,008.51	$80.00	4.1	$82.26	-$2.26
5	$2,010.85	$80.00	4.1	$82.35	-$2.35
6	$2,013.30	$80.00	4.1	$82.45	-$2.45
7	$2,015.84	$80.00	4.1	$82.55	-$2.55
8	$2,018.49	$80.00	4.1	$82.65	-$2.65
9	$2,021.25	$80.00	4.1	$82.76	-$2.76
10	$2,024.12	$80.00	4.1	$82.87	-$2.87
11	$2,027.11	$80.00	4.1	$82.99	-$2.99
12	$2,030.22	$80.00	4.1	$83.11	-$3.11
13	$2,033.46	$80.00	4.1	$83.24	-$3.24
14	$2,036.84	$80.00	4.1	$83.37	-$3.37
15	$2,040.35	$80.00	4.1	$83.51	-$3.51
16	$2,044.00	$80.00	4.1	$83.65	-$3.65
17	$2,047.80	$80.00	4.1	$83.80	-$3.80
18	$2,051.76	$80.00	4.1	$83.96	-$3.96
19	$2,055.89	$80.00	4.1	$84.12	-$4.12
...	...	...		...	...

Deliverables

1) You must upload all of your code to your assignment github folder.
2) Your code must include a makefile that will correctly compile your program.
3) You must create a Jenkins job that will poll your github repo hourly looking for changes in the repo. When triggered, it should clone your repo locally, call your makefile to compile your code, and it should run your program.
    a. I have added code to main.cpp that will allow you to pass in arguments from the command line when you call your file. You should run your file and pass it in the 3 values you need to calculate the results.

       Example: ./a.out 1000 18 50
4) The TA will compare your output to specific test cases we have defined. The output should match.
    a. You should call your executable file 14 times, passing it each of the test cases so that the TA can see all of the output.

       Example:

       ./a.out 1000 18 50
       ./a.out 2000 12 80
       ./a.out 1000 12 11
       ./a.out 1000 12 1010
       … etc