**Problem #1**

Subject: Mortgage Amortization Schedule

Filename: finance.py
Allowed modules: NONE

**WORK**

Most of the work for this assignment is already completed in HW01 and HW02. After fixing the errors I made in HW01 and HW02, I will copy over the mortgage_residual and mortgage_payment functions. To complete the mortgage_amortization function, I will first set up the header, title, and summary to get them out of the way. For the table, I need to first run through every month to collect the interest and principle of each month, and a running total of the interest, principle, and collective amounts paid. I will then have to something within this loop to print each month specified with the increment.

**OUTPUT**

```
DATA ENTRY

Enter loan amount ($):...... 480000

Enter loan APR (%):......... 7

Enter loan term (yr):....... 30

Filename (w/o ext):......... 480_30_7000


MORTGAGE AMORTIZATION SCHEDULE


Loan amount: ......$480000.0

Loan rate: ........ 7.0%

Loan term: ........ 30 years

Monthly Payment: ..$3193.45

Final Payment: ....$3196.11

Total Paid: .......$1149644.66

Cost of Credit: ...$669644.66
```

```
      /----------Payment------------\ /-----------Total-------------\
month   payment  interest principle  interest principle     paid   balance
   12   3193.45   2774.00    419.45  33445.54   4875.86  38321.40 475124.14
   24   3193.45   2743.68    449.77  66538.60  10104.20  76642.80 469895.80
   36   3193.45   2711.17    482.28  99253.72  15710.48 114964.20 464289.52
   48   3193.45   2676.30    517.15 131563.56  21722.04 153285.60 458277.96
   60   3193.45   2638.92    554.53 163438.79  28168.21 191607.00 451831.79
   72   3193.45   2598.83    594.62 194848.06  35080.34 229928.40 444919.66
   84   3193.45   2555.85    637.60 225757.65  42492.15 268249.80 437507.85
   96   3193.45   2509.76    683.69 256131.42  50439.78 306571.20 429560.22
  108   3193.45   2460.33    733.12 285930.66  58961.94 344892.60 421038.06
  120   3193.45   2407.33    786.12 315113.84  68100.16 383214.00 411899.84
  132   3193.45   2350.51    842.94 343636.44  77898.96 421535.40 402101.04
  144   3193.45   2289.57    903.88 371450.66  88406.14 459856.80 391593.86
  156   3193.45   2224.23    969.22 398505.31  99672.89 498178.20 380327.11
  168   3193.45   2154.16   1039.29 424745.47 111754.13 536499.60 368245.87
  180   3193.45   2079.03   1114.42 450112.29 124708.71 574821.00 355291.29
  192   3193.45   1998.47   1194.98 474542.61 138599.79 613142.40 341400.21
  204   3193.45   1912.09   1281.36 497968.75 153495.05 651463.80 326504.95
  216   3193.45   1819.46   1373.99 520318.12 169467.08 689785.20 310532.92
  228   3193.45   1720.13   1473.32 541512.87 186593.73 728106.60 293406.27
  240   3193.45   1613.62   1579.83 561469.54 204958.46 766428.00 275041.54
  252   3193.45   1499.42   1694.03 580098.63 224650.77 804749.40 255349.23
  264   3193.45   1376.96   1816.49 597304.15 245766.65 843070.80 234233.35
  276   3193.45   1245.64   1947.81 612983.19 268409.01 881392.20 211590.99
  288   3193.45   1104.84   2088.61 627025.42 292688.18 919713.60 187311.82
  300   3193.45    953.85   2239.60 639312.49 318722.51 958035.00 161277.49
  312   3193.45    791.95   2401.50 649717.55 346638.85 996356.40 133361.15
```

```
 324   3193.45    618.34   2575.11  658104.53  376573.27 1034677.80  103426.73
 336   3193.45    432.19   2761.26  664327.55  408671.65 1072999.20   71328.35
 348   3193.45    232.58   2960.87  668230.17  443090.43 1111320.60   36909.57
 360   3196.11     18.54   3177.57  669644.66  480000.00 1149644.66      -0.00
```

```
DATA ENTRY
Enter loan amount ($):...... 480000
Enter loan APR (%):......... 6.875
Enter loan term (yr):....... 15
Filename (w/o ext):......... 480_15_6875

MORTGAGE AMORTIZATION SCHEDULE


Loan amount: ......$480000.0
Loan rate: ........ 6.875%
Loan term: ........ 15 years
Monthly Payment: ..$4280.90
Final Payment: ....$4281.16
Total Paid: .......$770562.26
Cost of Credit: ...$290562.26


      /----------Payment------------\ /-----------Total-------------\
 month   payment  interest  principle  interest  principle      paid   balance
   12   4280.90   2650.71   1630.19   32409.94   18960.86   51370.80  461039.14
   24   4280.90   2535.03   1745.87   63474.44   39267.16  102741.60  440732.84
   36   4280.90   2411.15   1869.75   93098.02   61014.38  154112.40  418985.62
   48   4280.90   2278.48   2002.42  121178.47   84304.73  205483.20  395695.27
   60   4280.90   2136.39   2144.51  147606.28  109247.72  256854.00  370752.28
   72   4280.90   1984.22   2296.68  172264.18  135960.62  308224.80  344039.38
```

```
  84   4280.90   1821.25   2459.65  195026.59  164569.01  359595.60  315430.99
  96   4280.90   1646.72   2634.18  215759.00  195207.40  410966.40  284792.60
 108   4280.90   1459.80   2821.10  234317.36  228019.84  462337.20  251980.16
 120   4280.90   1259.62   3021.28  250547.40  263160.60  513708.00  216839.40
 132   4280.90   1045.23   3235.67  264283.90  300794.90  565078.80  179205.10
 144   4280.90    815.64   3465.26  275349.95  341099.65  616449.60  138900.35
 156   4280.90    569.75   3711.15  283556.06  384264.34  667820.40   95735.66
 168   4280.90    306.41   3974.49  288699.27  430491.93  719191.20   49508.07
 180   4281.16     24.39   4256.77  290562.26  480000.00  770562.26     -0.00
```

**CODE**

```
'''
PROGRAMMER: Christopher Colbert
USERNAME: ccolbert
PROGRAM: finance.py

DESCRIPTION: Mortgage Functions
'''


def mortgage_residual(amount, rate, term, payment):
    #Convert APR to monthly interest
    monthly_rate= (rate/100)/12

    remaining_balance=amount

    #for each month
    for month in range(0, term*12):
        #subtracting interest from payment
        amount= round(remaining_balance * monthly_rate, 2)
        principal_payment = payment - amount

        #take payment - interest out of balance
        remaining_balance -= principal_payment

    return remaining_balance
#end mortgage_residual function

def mortgage_payment(amount, rate, term):
    #Convert APR to monthly interest
    monthly_rate= (rate/100)/12

    monthly_payment = round(amount * (monthly_rate * (1 + monthly_rate) ** (term * 12)) / ((1 + monthly_rate) ** (term * 12) - 1),2)

    final_payment = round(monthly_payment + mortgage_residual(amount, rate, term, monthly_payment),2)

    return monthly_payment, final_payment
#end mortgage_payment function



def mortgage_amortization(amount, rate, term, increment=1):
    monthly_rate = rate/100/12

    monthly_payment, final_payment = mortgage_payment(amount,rate,term)

    total_paid = 0
    total_interest = 0
```

```
  title = "MORTGAGE AMORTIZATION SCHEDULE\n"


  header = ("\n%8s%33s%33s" %("", r"/----------Payment------------\ ", r" /-----------Total-------
------\ "))
  header += ("\n%6s%11s%11s%11s%11s%11s%11s%11s"
       %("month","payment","interest","principle","interest","principle","paid","balance"))
  table = ""
  remaining_balance = amount


  #for each month
  for month in range(0, term * 12 - 1):
    interest_payment = round(remaining_balance * monthly_rate,2)
    principal_payment = monthly_payment - interest_payment
    total_interest += interest_payment
    total_paid += monthly_payment
    remaining_balance -= principal_payment


    #add every increment month to table
    if (month + 1) % increment == 0:
      table += f"{month+1:6} {monthly_payment:10.2f} {interest_payment:10.2f}
{principal_payment:10.2f} {total_interest:10.2f} {total_paid - total_interest:10.2f}
{total_paid:10.2f} {remaining_balance:10.2f}\n"



  # Add final payment
  interest_payment = remaining_balance * monthly_rate
  principal_payment = final_payment - interest_payment
  total_interest += interest_payment
  total_paid += final_payment
```

```
    remaining_balance -= principal_payment

    table += f"{term * 12:6} {final_payment:10.2f} {interest_payment:10.2f}
{principal_payment:10.2f} {total_interest:10.2f} {total_paid - total_interest:10.2f}
{total_paid:10.2f} {remaining_balance:10.2f}\n"

    #format summary string
    summary = f"Loan amount: ......${amount}\n"
    summary += f"Loan rate: ........ {rate}% \n"
    summary += f"Loan term: ........ {term} years\n"
    summary += f"Monthly Payment: ..${monthly_payment:.2f}\n"
    summary += f"Final Payment: ....${final_payment:.2f}\n"
    summary += f"Total Paid: .......${total_paid:.2f}\n"
    summary += f"Cost of Credit: ...${total_interest:.2f}"

    return (title,summary,header,table)

def mortgage_report(amount, rate, years, increment = 1):

    report = ""
    for s in mortgage_amortization(amount, rate, years, increment):
        report += s
    return report
```

**Problem #2**

Subject:

Filename: hw03_02.py
Allowed modules: none

**WORK**

To solve this problem, I will first copy over hw03_01.py and finance.py into a separate code folder to preserve the solution from problem 1. I will then adjust finance to change the given payment to support accelerated payoff. I will need to add two additional parameters, one to add an additional percent of the original payment to the monthly payment, and the other to add an additional percent of the principal payment to the monthly payment. I will then add these parameters into the running calculation. The last thing left to do will then be to add a section in the summary showing how many much time and money was saved by using the accelerated payment schedule.

**OUTPUT**

```
DATA ENTRY

Enter loan amount ($):...... 480000

Enter loan APR (%):......... 7

Enter loan term (yr):....... 30

Filename (w/o ext):......... 480_30_7000_500extra


MORTGAGE AMORTIZATION SCHEDULE


Loan amount: ..........$480000.0

Loan rate: ............ 7.0%

Loan term: ............ 30 years

Monthly Payment: ......$3193.45

Final Payment: ........$3639.31

Total Paid: ...........$901170.57

Cost of Credit: .......$421191.80

Accelerated Payoff:
```

```
Time Saved: ........... 9 years 8 months

Amount Saved: .........$248917.29238999565

Interest Saved: .......$248452.8565027247

Total Saved:...........$497370.14889272023


     /----------Payment------------\ /-----------Total-------------\
 month   payment  interest  principle  interest  principle     paid   balance
   12   3193.45   2740.96    952.58  33249.21  11073.32  44322.53  468926.68
   24   3193.45   2672.10   1021.44  65697.92  22947.14  88645.06  457052.86
   36   3193.45   2598.26   1095.28  97288.28  35679.31 132967.59  444320.69
   48   3193.45   2519.08   1174.46 127958.22  49331.90 177290.12  430668.10
   60   3193.45   2434.18   1259.36 157641.21  63971.45 221612.66  416028.55
   72   3193.45   2343.14   1350.40 186265.92  79669.27 265935.19  400330.73
   84   3193.45   2245.52   1448.02 213755.83  96501.89 310257.72  383498.11
   96   3193.45   2140.84   1552.70 240028.92 114551.33 354580.25  365448.67
  108   3193.45   2028.60   1664.94 264997.20 133905.58 398902.78  346094.42
  120   3193.45   1908.24   1785.30 288566.37 154658.94 443225.31  325341.06
  132   3193.45   1779.18   1914.36 310635.26 176912.58 487547.84  303087.42
  144   3193.45   1640.79   2052.75 331095.46 200774.91 531870.37  279225.09
  156   3193.45   1492.39   2201.15 349830.62 226362.29 576192.91  253637.71
  168   3193.45   1333.27   2360.27 366716.07 253799.37 620515.44  226200.63
  180   3193.45   1162.65   2530.89 381618.12 283219.85 664837.97  196780.15
  192   3193.45    979.69   2713.85 394393.33 314767.17 709160.50  165232.83
  204   3193.45    783.50   2910.04 404888.01 348595.02 753483.03  131404.98
  216   3193.45    573.14   3120.40 412937.26 384868.30 797805.56   95131.70
  228   3193.45    347.56   3345.98 418364.31 423763.78 842128.09   56236.22
  240   3193.45    105.68   3587.86 420979.59 465471.03 886450.62   14528.97
  244   3639.31      0.00   3693.54 421191.80 479978.77 901170.57       0.00
```

```
DATA ENTRY
Enter loan amount ($):...... 480000
Enter loan APR (%):......... 7
Enter loan term (yr):....... 30
Filename (w/o ext):......... 480_30_7000_double_principal

MORTGAGE AMORTIZATION SCHEDULE


Loan amount: ..........$480000.0
Loan rate: ............ 7.0%
Loan term: ............ 30 years
Monthly Payment: ......$3193.45
Final Payment: ........$3307.08
Total Paid: ...........$816474.54
Cost of Credit: .......$336493.83
Accelerated Payoff:
Time Saved: ........... 14 years 11 months
Amount Saved: .........$333281.0899999923
Interest Saved: .......$333150.82452499954
Total Saved:...........$666431.9145249919


     /----------Payment------------\ /-----------Total-------------\
 month   payment  interest  principle   interest  principle     paid   balance
   12   3193.45   2746.46    893.98   33284.96   10072.88   43357.84  469927.12
   24   3193.45   2679.70   1027.50   65817.76   21650.08   87467.84  458349.92
   36   3193.45   2602.98   1180.94   97486.08   34956.24  132442.32  445043.76
   48   3193.45   2514.80   1357.30  128160.81   50249.58  178410.39  429750.42
   60   3193.45   2413.44   1560.02  157693.57   67826.86  225520.43  412173.14
```

```
  72   3193.45   2296.96   1792.98  185913.81   88029.18  273942.99  391970.82
  84   3193.45   2163.07   2060.76  212625.54  111248.52  323874.06  368751.48
  96   3193.45   2009.19   2368.52  237603.45  137935.50  375538.95  342064.50
 108   3193.45   1832.33   2722.24  260588.62  168607.96  429196.58  311392.04
 120   3193.45   1629.06   3128.78  281283.43  203861.14  485144.57  276138.86
 132   3193.45   1395.43   3596.04  299345.86  244379.08  543724.94  235620.92
 144   3193.45   1126.91   4133.08  314382.75  290948.10  605330.85  189051.90
 156   3193.45    818.29   4750.32  325942.30  344471.80  670414.10  135528.20
 168   3193.45    463.58   5459.74  333505.19  405988.82  739494.01   74011.18
 180   3193.45     55.90   6275.10  336474.54  476692.92  813167.46    3307.08
 181   3307.08      0.00   6386.90  336493.83  479980.71  816474.54       0.00
```

**CODE**

```
'''
PROGRAMMER: Christopher Colbert
USERNAME: ccolbert
PROGRAM: finance.py

DESCRIPTION: Mortgage Functions with accelerated payoff option.
'''


def mortgage_residual(amount, rate, term, payment):
  #Convert APR to monthly interest
  monthly_rate= (rate/100)/12

  remaining_balance=amount
```

```
 #for each month
    for month in range(0, term*12):
        #subtracting interest from payment
        amount= round(remaining_balance * monthly_rate, 2)
        principal_payment = payment - amount

        #take payment - interest out of balance
        remaining_balance -= principal_payment

    return remaining_balance
#end mortgage_residual function

def mortgage_payment(amount, rate, term):
    #Convert APR to monthly interest
    monthly_rate= (rate/100)/12

    monthly_payment = round(amount * (monthly_rate * (1 + monthly_rate) ** (term * 12)) /
((1 + monthly_rate) ** (term * 12) - 1),2)

    final_payment = round(monthly_payment + mortgage_residual(amount, rate, term,
monthly_payment),2)

    return monthly_payment, final_payment
#end mortgage_payment function
```

```
def mortgage_amortization(amount, rate, term, fixed = 0, variable = 0, increment = 1):
  term_months = term*12
  monthly_rate = rate/100/12

  monthly_payment, final_payment = mortgage_payment(amount,rate,term)

  #initialize totals
  total_paid = 0
  total_interest = 0
  final_month = 0

  title = "MORTGAGE AMORTIZATION SCHEDULE\n"

  header = ("\n%8s%33s%33s" %("", r"/----------Payment------------\ ", r" /-----------Total-------------\ "))
  header += ("\n%6s%11s%11s%11s%11s%11s%11s%11s"
        %("month","payment","interest","principle","interest","principle","paid","balance"))
  table = ""
  remaining_balance = amount

  #for each month calculate total paid, total interest,
  for month in range(term_months):
    interest_payment = round(remaining_balance * monthly_rate,2)
    principal_payment = monthly_payment - interest_payment

    principal_payment += variable / 100 * principal_payment

    principal_payment += fixed / 100 * monthly_payment
    #check if principal exceeds remaining balance
    if remaining_balance > principal_payment:
      total_interest += interest_payment
      total_paid += monthly_payment + (fixed / 100 * monthly_payment) + (variable / 100 *
(monthly_payment - interest_payment))
      remaining_balance -= principal_payment


      #add every incremented month to table
      if (month + 1) % increment == 0:
        table += f"{month+1:6} {monthly_payment:10.2f} {interest_payment:10.2f}
{principal_payment:10.2f} {total_interest:10.2f} {total_paid - total_interest:10.2f}
{total_paid:10.2f} {remaining_balance:10.2f}\n"
    #if principal is greater than remaining, save last month, final payment,
    else:
      if final_month == 0:
        final_month = month + 1


        # Add final payment for normal payoff option
        interest_payment = remaining_balance * monthly_rate
        final_payment = remaining_balance
        principal_payment = final_payment - interest_payment
        total_interest += interest_payment
        total_paid += final_payment
        remaining_balance -= final_payment
  if final_month == 0:
    table += f"{term*12:6} {final_payment:10.2f} {interest_payment:10.2f}
{principal_payment:10.2f} {total_interest:10.2f} {total_paid - total_interest:10.2f}
{total_paid:10.2f} {remaining_balance:10.2f}\n"
```

```
else:
        table += f"{final_month:6} {final_payment:10.2f} {interest_payment:10.2f} {principal_payment:10.2f} {total_interest:10.2f} {total_paid - total_interest:10.2f} {total_paid:10.2f} {remaining_balance:10.2f}\n"


    #format summary string
    summary = f"Loan amount: ..........${amount}\n"
    summary += f"Loan rate: ............ {rate}% \n"
    summary += f"Loan term: ............ {term} years\n"
    summary += f"Monthly Payment: ......${monthly_payment:.2f}\n"
    summary += f"Final Payment: ........${final_payment:.2f}\n"
    summary += f"Total Paid: ...........${total_paid:.2f}\n"
    summary += f"Cost of Credit: .......${total_interest:.2f}"


    #if accelerated, show time and money saved from original
    if fixed > 0 or variable >0:
        #get original total and interest total
        original_total = 0
        original_interest = 0
        remaining_balance = amount


        for month in range(0, term * 12 - 1):
            interest_payment = round(remaining_balance * monthly_rate,2)
            principal_payment = monthly_payment - interest_payment
            original_interest += interest_payment
            original_total += monthly_payment
            remaining_balance -= principal_payment


        interest_payment = remaining_balance * monthly_rate
        principal_payment = final_payment - interest_payment
        original_interest += interest_payment
        original_total += final_payment
        remaining_balance -= principal_payment



        accelerated_years = int(term - final_month / 12)
        accelerated_months = term * 12 - final_month - accelerated_years * 12
        summary += f"\nAccelerated Payoff:"
        summary += f"\nTime Saved: ........... {accelerated_years} years {accelerated_months} months"
        summary += f"\nAmount Saved: .........${original_total - total_paid}"
        summary += f"\nInterest Saved: .......${original_interest - total_interest}"
        summary += f"\nTotal Saved:...........${original_total + original_interest - total_interest - total_paid}"

    return (title,summary,header,table)

def mortgage_report(amount, rate, years, increment = 1, fixed = 0, variable = 0):
    report = ""
    for s in mortgage_amortization(amount, rate, years, increment, fixed, variable):
        report += s
    return report
```