Homework 1

Task

Write a program for loan amortization swapping. Consider an n1-year, V-dollar loan at an r1 interest rate. At the end of year n2 (n2 < n1), however, there is an opportunity to swap the loan (i.e., the remaining principal) into a new loan with an r2 interest rate for the remaining n1-n2 years, of course under a new amortization. An F-dollar fee is charged at the end of year n2 if the loan is swapped. Both loans have the same number of payments per annum, m.

Inputs

- V: original loan amount in dollars
- m: number of payments per annum, an integer
- n1: duration of the original loan in years, an integer
- r1: annual interest rate of the original loan, compounded m times per annum
- n2: the year that loan swapping is possible
- r2: annual interest rate of the new loan, also compounded m times per annum
- F: swapping fee

Outputs

- Total principal paid in the first n2 years
- Total interest paid in the first n2 years
- Total interest paid from the end of year n2 (excluded) to the end of year n1 if the loan is not swapped (so the r1 interest rate is maintained)
- Total interest paid from the end of year n2 (excluded) to the end of year n1 if the loan is swapped to the new r2 interest rate
- The IRR for the whole n1 years (the fee F considered) if the loan is swapped
- Does swapping the loan lower the IRR? Answer 1 for yes, 0 for no difference, and -1 for no

Example

If V = 1000000, m = 12, n1 = 2, r1 = 0.060, n2 = 1, r2 = 0.025, F = 888, the outputs are 485041.840313, 46805.482720, 16889.163346, 6999.998244, 0.051895.

- Input format (for Python codes):
  “python3 (your_file_name).py 1000000 12 2 0.060 1 0.025 888”
- Output format:
  "485041.840313, 46805.482720, 16889.163346, 6999.998244, 0.051895, 1"

Supplementary information

1. The IRR should be annualized. Both continuous annualized and discrete annualized (based on the number of payments per annum) are acceptable.
2. If you use continuous annualization, the annualized IRR of the example output would be 0.053147 or 0.053265. All three answers, 0.051895, 0.053147, and 0.053265 are acceptable.

Private testcases (released after the deadline)

1. Inputs: 1000000 12 2 0.060 1 0.025 888
   Outputs: 485041.840313 46805.482720 16889.163346 6999.998244 0.051895 (or 0.053147 or 0.053265) 1
2. Inputs: 9999999 4 4 0.045 2 0.045 0
   Outputs: 4776404.960369 715084.514797 267895.435535 267895.435535 0.045000 (or 0.045765 or 0.046028) 0
3. Inputs: 8888888.888 2 10 0.150 9 0.149 222222.222
   Outputs: 7323280.685672 8371469.973932 178252.982072 177051.080776 0.151763 (or 0.157521 or 0.163884) -1
4. Inputs: 7777777 6 30 0.015 29 0.014 66666
   Outputs: 7458306.780901 1887563.816274 2801.180803 2614.073932 0.015394 (or 0.015394 or 0.015514) -1
5. Inputs: 666666 12 20 0.6 1 0.5 88888
   Outputs: 4.358276 399998.527450 6933393.187060 5667198.745200 0.587840 (or 0.775169 or 0.800097) 1