

**Background story:**

Imagine, One person is going to invest  $X$  \$. There are two options:

- put money on the deposit at any well-known bank and have  $r$  % annual return which equals to the interest rate in that country.
- lend money to a new growing business and have  $R$  % annual return ( $R > r$ ).

Second option brings higher return but involves taking some risk in case of business bankruptcy.

After investigating information about the start-up business and its founders, the person has figured out that default of that growing company is very unlikely, so it's really a good opportunity to make such investment to have extra return comparing to the market interest rate.

So, the second option has been chosen.

**Investment details:**

1. Interest rate is  $R$  % per year.
2. A borrower should return his debt as fixed payment each month. Monthly payment doesn't change each month and represents the sum of two components: part of initial principal and interest amount.
3. Interest amount is calculated on the outstanding principal amount.
4. Investment duration is  $N$  years (it means last refund payment should be done in  $N$  years after making the investment)

**The task:**

Write a module (i.e. several functions with same prefix) that takes: **Agreement date, Calculation date,  $X$ ,  $R$  and  $N$**  as input data and calculates Sum of all future interest payments.

Code requirements/restrictions:

- You can use any programming language you familiar with (but APL/C# is a plus)
- **The code should follow maintainability principals.**
- There should be well defined single-entry point (interface) in the code that takes input argument and returns the result.