Function **value\_abc:**

1. Calculated total number of days **without** using the datetime module of python.
2. **get\_diff() function**: assigned day number to both the dates and returned the difference of two.
3. tau() function: class member, returns difference in total number of months
4. Calculated N by tau() function and freq value
5. Calculated cf\_i using the formula given in question. Stored the values in **cf[] array**
6. Initialized period\_start\_i with t\_start and then iterated over N periods. Calculated period\_end\_i using newdate() function and freq value
7. **newdate()** **function** : Since format of dates is not very common . I have made this function which takes one date and freq value as argument (in format specified in question) and returns the next date (period end)
8. Calculated mu\_N using the formula given in Question. Since mu\_N depends on its previous(period) value. So, I made a mu[] array which iteratively calculates the mu\_0 to mu\_N. Here I have also handled the **edge case** of tdeal and period\_start and prd.
9. As all the variables are calculated to be used in calculating vdeal. In the end, vdeal is calculated using the formula given in Question. And vdeal is returned.

Function **rateofreturn\_abc**

1. Ordeal is dependent on vdeal and mu\_N. mu\_N is further dependent on rdeal. Infact there is a ongoing series till mu\_0.
2. This will lead to a equation of ordeal in power of N.
3. So, We start with rdeal =0 , and calculate the value of vdeal\_cal using this rdeal and tdeal from the known method of calculating value\_abc. Now we calculate the difference between actual vdeal and vdeal\_cal. The value of rdeal at which this difference is minimised we say that it is our answer.
4. We kept our increment as 0.01

Function **dvalue\_drateofreturn\_abc**

1. See the formula in picture attached.
2. Apart from the things we are calculating in **value\_abc()** function this time we also need to calculate dmu\_i\_dr for each i
3. So while calculating mu[] array , we also construct our dmu\_dr[] array using the time(i) series formula (on mu\_i) given in the question. Which store the differentiation of mu[i] w.r.t rdeal
4. Finally using the formula shown in picture , I calculate the dv\_dr.

Function **dvalue\_drateofreturn\_abc**

1. See the formula in picture attached.
2. Apart from the things we are calculating in **dvalue\_drateofreturn\_abc()** function this time we also need to calculate d2mu\_i\_dr2 for each i
3. So while calculating mu[] array and dmu\_dr[] array , we also construct our d2mu\_dr2[] array using the time(i) series formula (on mu\_i) given in the question. Which store the double differentiation of mu[i] w.r.t rdeal
4. Finally using the formula shown in picture , I calculate the d2v\_dr2.

If you cannot see the below picture properly , use this drive link to see the picture.

<https://drive.google.com/file/d/1mCtzCUsp6VTsJU6Zoh3ueahItm8tTMgA/view?usp=sharing>

