

Lab Assignment 3: Tuition savings calculator

With the help of an RESP (Registered Education Savings Plan), a parent can start putting aside money for a child's post-secondary education. Suppose that a young couple deposits **an initial amount** to start the savings plan, and each month thereafter they contribute some monthly fixed amount. Assume that the savings amount will grow at a fixed annual interest rate compounded monthly.

Each month the balance increases according to the following formula:

$$\begin{aligned} \text{New balance} = & \text{Old balance} + (\text{Old balance} * \text{Monthly interest rate}) \\ & + \text{Monthly contribution} \end{aligned} \quad (1)$$

Since the university tuition fee increases over the years, you are asked to further analyze whether the saving is enough to cover the tuition fee for a 4-year study program when the child turns 18 and enters post-secondary education. You will help them analyze this for 3 different types University programs: Arts, Science, and Engineering, for which the average tuition fees in the current year are as follows: **\$5550**, **\$6150**, and **\$6550** respectively. The predicted average percentage of tuition increase *each year* is **7%**.

Each year the new fee due to the annual increase is calculated as follows:

$$\text{New cost} = \text{Old cost} + (\text{Old cost} * \text{Annual increase rate}) \quad (2)$$

Version 0: Getting Started

Download and unzip the files in the **V0GetStarted.zip** file into your Working Directory. Open the lab3_v0.py file in the Editor Window and inspect it before running it. The program simply generates a counter variable that increments by 1 and stores it as a vector using a **for loop**. A simple linear plot is also produced. The console output and plot should match the ones provided in the V0GetStarted zip file.

Version 1: Savings/Tuition Calculation

Download and unzip the files in the **V1Saving.zip** file into your Working Directory. In Version 1, you are asked to calculate the balance of the saving account in each month over 18 years, so the total saving by the end of the 18th year is known. Save the result as a list/ndarray type. Assume the parents deposited an initial amount of \$2000 with further future monthly contributions of \$200. The savings will grow at an interest rate of 6.25% per year compounded monthly. A **for** loop must be used for the above calculation. Add a user input prompt to allow the user to select which University program tuition cost will be displayed.

Input	1	2	3
Program	Arts	Science	Engineering

Calculate the *predicted total* tuition fee for the program selected by the start of the 18th year, which should be the total fee over a 4-year university program (i.e. from year 18 to year 21). A ~~for~~ loop must be used for the above calculation. Print the final savings amount and the program tuition cost to 2 decimal spots in the console and display them graphically through a plot. The plot should contain the annual savings with respect to the number of years and a horizontal line (threshold) indicating the tuition fee for the program selected. Annotate the graph by adding a suitable title, label all axis and a provide legend. Please modify the provide lab3_v0.py file to write your program. You may want to create a backup copy before writing your code. Remove or modify the exiting code to satisfy the above requirements. Before submitting, please rename the file to include your CCID.

The output in the console window should match the storyboard in the lab3Output_v1.txt file. Your plot should also match the one in lab3Plot_v1. Include a section comment header called `User Input, Savings/Tuition Calculation, Plot, and Print Output` and summarize what the sections does.

Version 2: Customization

Download and unzip the files in the **V2Customization.zip** file into your Working Directory. To increase the flexibility of the savings program, before prompting the user to select the University program, allow them to enter an **initial deposit, the monthly contribution and the interest rate** that the savings will increase by in that order. Using the same savings formula in version 1, determine if the new savings balance satisfies the program tuition cost selected. Display in the console window the savings amount to 2 decimals places and whether it is sufficient or not along meet the program cost. Finally, if the savings amount is not sufficient, you will determine at what minimum monthly contribution amount (to the **nearest dollar**) would be required in order to satisfy the program tuition cost. You must perform this calculation with a **while** loop. Assume that the monthly amount which is fixed for the duration of the calculation, starts at 1 dollar and will increment by 1 dollar each time the savings amount does not meet the program tuition cost. Finally, **update the plot to include the new saving balance**. Before submitting, please rename the file to include your CCID.

The output in the console window should match the storyboard in the lab3Output_v2.txt file for the given test case. Along with the section comment headers from version 1, add an additional section comment header called `Optimal Monthly Amount` and summarize what that section does.