Project 1 - Financial Literacy

Inspired by: the Freakonomics podcast *Everything You Always Wanted to Know About Money (But Were Afraid to Ask)*
and discussion with your classmate Ilana Seidl.

Project is due Wednesday, April 10 at 11:59 p.m. (one deadline)

Background

Overarching Background

"If you've spent any time reading up on education or financial news lately, you've probably come across the term *financial literacy*. The goal behind teaching financial literacy is to help people develop a stronger understanding of basic financial concepts—that way, they can handle their money better.

That's a worthy goal, especially when you consider a few stats about how the typical American handles money:(1)

- Nearly four out of every five U.S. workers live paycheck to paycheck.
- Over a quarter never save any money from month to month.
- Almost 75% are in some form of debt, and most assume they always will be.

With those numbers, it's no surprise that leaders in business, education and government want to help spread the benefits of greater financial literacy to as many people as possible.

So, what is financial literacy?
Financial literacy is the possession of skills that allow people to make smart decisions with their money.
And don't be misled by the word *literacy*. Although understanding stats and facts about money is great, no one has truly grasped financial literacy until they can regularly make smart decisions with their money that leads to positive financial outcomes.
When you have this skill set, you're able to understand the major financial issues most people face: emergencies, debts, investments, and beyond. Financially literate people know their way around a budget, know how to use sinking funds, and can tell you the difference between a 401(k) and 529" (What do you need to know about financial literacy?).

Test your financial literacy

Want to test your financial literacy? Answer these questions and see how well you did here.

1) “Suppose you had $100 in a savings account and the interest rate was 2% per year. After 5 years, how much do you think you would have in the account if you left the money to grow?”
A) More than $102
B) Exactly $102
C) Less than $102
D) Don’t know

2) “Imagine that the interest rate on your savings account was 1% per year and inflation was 2% per year. After 1 year, with the money in this account, would you be able to buy...”
A) More than today
B) Exactly the same as today
C) Less than today
D) Don’t know

3) “Do you think the following statement is true or false?
Buying a single company stock usually provides a safer return than a stock mutual fund.”
A) True
B) False
C) Don’t know

If you didn’t get all three questions right, you’re not alone. This quiz has been used across many countries and demographics, and the majority do not correctly answer all three questions. Notably, “surveys have demonstrated generally lower levels of financial literacy among women, those with low levels of education, and certain minority groups such as African-Americans and Hispanics (Lusardi 11-12). Such barriers to financial literacy are not a matter of intelligence. Instead, traditional gender roles regarding accumulating and managing wealth, socio-economic factors such as income and access to education, cultural attitudes on consumption and financial services, and language barriers often affect the development of financial literacy within those demographics. (Simulations for Financial Literacy)”

If you'd like to read more about how various demographics did on this survey, you can read more here. If you'd like to test your own financial literacy in a game/simulation where you earn minimum wage, click here. (It's very hard.)

Assignment Details

Introduction

In this assignment you're going to be simulating how much money a financially literate versus a non-financially literate person has after 40 years. They both start out with the same amount of money, the same amount of debt, earn the same amount, and buy houses that cost the same amount. The small decisions they make end up accumulating, and by the end of the simulation, you'll see that there's a significant disparity between how much wealth the financially literate and the non-financially literate have.

Financial literacy encompasses a huge amount of knowledge and information. This simulation is *vastly simplifying* the use of money in life in order to make the assignment more manageable and to allow you to see exactly how certain financial decisions have significant outcomes. The three financial literacy questions that you will be exploring through this assignment are as follows:

- Should you put your savings in a bank savings account or in a mutual fund?
- How much of a difference does it make if you pay off your credit card debt now or later?
- Should you wait until you can afford a 20% down payment before you buy a house?

Almost every function that you write will take in a pointer to struct as a parameter. For simplicity of explanation, I will call these two possible structs `fl` and `nfl, fl` (Financially Literate) representing the accounts of the financially literate person and `nfl` (Not Financially Literate) representing the accounts of the person who is not financially literate. Each struct includes starting financial info for `fl` and `nfl`. They both have a "savings" account with a corresponding number that represents the amount in savings they have, a "checking" account with money that they use to pay mortgage, "debt" from their credit cards, a "loan" for when they take out a loan for their house, "yearsWithDebt" that will show the number of years each one remains in debt, "yearsRented" that will show the number of years that person rented, and "debt paid" will show the total amount that the person paid to pay off the $30,100 debt.

*Coincidentally, most players on the National Football League are **N**ot **F**inancially **L**iterate. 78% of professional football players go bankrupt or are under financial stress within 2 years of retiring, despite having made millions over their careers. You can read more here or listen to a Freakonomics podcast about it called *"Everything You Always Wanted to Know About Money (But Were Afraid to Ask)"*.

Savings Placement

```
savingsPlacement(person, interest rate)
```

This function takes in a struct (which will be either fl or nfl) and yearly interest rate and modifies that struct to reflect that person's savings after one year.-`fl` and `nfl` both start off with $5000 that they'd like to save. `nfl` puts the money in a simple savings account, which (like most saving accounts) has a 1% annual interest. `nfl` doesn't know that since inflation is 2% per year, his money will be worth less and less every year. `fl` knows about inflation, so they put their money in a mutual fund. Mutual funds typically have an annual interest rate of 7%.

- Write a function that updates the amount of money each one has in their bank accounts after one year. nfl's grows 1% each year, and fl's grows 7%. The only value in the struct that should change is the "savings" account.

Paying Off Credit Card Debt

```
debt(person, ~~initial amount,~~ interest rate, addl pay)
```

`fl` and `nfl` both have $30,100 of debt on their credit cards and in student loans. Since the interest rates are similar for both, we've combined them under the single key called `"debt"`. Each month, the minimum amount that the credit card company requires be paid off is 3%. `nfl` pays off just the minimum 3% each month plus an additional dollar (the additional dollar is important because without it, the function would continue taking 3% of the remaining debt, and consequently would approach 0 but never reach it). `fl` pays off the minimum 3% plus an additional $15 of it, knowing that paying the debt off more quickly will reduce the amount that they ultimately end up paying. The interest on the debt that hasn't been paid off is 20% compounded annually. This is calculated each year by multiplying the remaining debt by 1.2.

- Write a function that calculates the amount of money left in debt and the amount of money they have left in their accounts for one year.

Details:

- The `"debtPaid"` key should be updated for the `person` each simulated month (not year).
- The simulation should stop on the *month* that the debt is no longer over $0.
- The `"yearsWithDebt"` key should be updated for the `person` each year.
- The function should take the parameters as specified in the signature.
- Since both `nfl` and `fl` haven't made enough money in their checking accounts yet to start paying off the debt, they use their savings accounts.

Different Down Payments on a House

`fl` and `nfl` decide that they'd like to buy houses. The difference is that they decide to buy them at different times in their lives. `nfl` buys the house as soon as they're able to afford a down payment of 5% on the house (so as soon as their checking is over 5% of the total cost of the house). `fl` buys the house as soon as they're able to afford a down payment of 20% on the house. Both are buying a house that costs $175,000, and the interest rate for both is 4.5%. What `nfl` doesn't know is that when you pay a down payment less than 20% on any house, an extra interest rate called "private mortgage insurance" (PMI) is added onto the cost. A 0.5% (assuming NFL has a great credit score) interest rate is added onto the amount that `nfl` has to pay each year.

Before each one moves into a house, they're living in apartments that cost them $850 a month (the average affordable amount in the US).

```
rent(person, rentamt)
```

Write a function that will calculate the amount each one will have left in their checking accounts after one year if they're in the rented apartment.
Details:

- The cost per month to rent is $850.
- The money comes out of the checking account.
- "yearsRented" should be updated.

```
house(person, houseprice, interestrate, mortgageterm)
```

Write a function that calculates the amount of money each one will have left in their checking accounts after one year. To calculate the monthly mortgage payment, you can assume that both `fl` and `nfl` will pay off their mortgages in full in 30 years and that they pay the same amount of money one year as they'll pay the next year.

Details:

- The interest on the loan for `fl` is 4.5%. The interest on the loan for `nfl` is 5% (that is 4.5% of loan and 0.5% of PMI).
- Update the "checking" account and "loan" left per month and create a loop that runs it for 12 months. *Do not try and calculate the total for the year without using the loop, because you will get a different amount.*
- To calculate the monthly mortgage payment
  - N (# payments) = 360 (30 years times 12 monthly payments per year)
  - i (interest) = (interest expressed in decimal format divided by 12)
  - D (discount factor) = [(1+i)^N-1] / (i(1+i)^N)
  - P (monthly payment) = LoanAmount / D
- The money comes from the checking account.

The formula is also explained here

(You can read more about buying a house with different down payments here.)

40-Year Simulation

```
simulator(person, yearlysalary):
```

For one year, these small choices that `fl` and `nfl` make may not have a huge effect on their savings and checking accounts, but over forty years you can see a much larger effect. Now you're going to be using all of the previous functions as helper functions to simulate how much `fl` and `nfl` will have after 40 years.
`fl` and `nfl` both earn $59,000 a year, the median of a middle-class person. They receive the money at the start of each year. They put 20% into savings, and 30% into checking. In writing the method, you can pretend that this 50% is all that they earn, since the other 40% is theoretically being used up elsewhere.
(15% is spent on food.
15% is spent on transportation.
10% is spent on miscellaneous.
So, the amount that ends up going into checking and saving at the end of each year is actually $29,500.)
Use a for loop and the helper functions to determine how much `fl` or `nfl` has left at the end of each year.
Details:

- `simulator(person, yearlysalary)` should return an array with 41 elements. Each element represents how much "wealth" the inputted person had that year. The initial

wealth (-24100) should be the first element in the array. These points will later be used in a graph.

- "Wealth" should be a single number that represents the total money the person has in their accounts (savings + checking - debt - loan), and it should be rounded to an int.
- Each year, the person should put 20% of their $59,000 annual income into savings, and 30% of their annual income into checking.
- Run `savingsPlacement( )` and `debt( )` for each simulated year.
- Before the person has enough money in their checking account to pay the 5% or 20% down payment, the person should pay rent each year. Once they have enough for the down payment, "loan" should be updated to be the price of the house minus the down payment, and then `house()` should be run each year until the loan has been paid off.
- ~~After the wealth has been calculated for that year, divide it by 3. This is to take taxes, emergencies, etc. into account, and to make the money they end up with more realistic.~~
- You should return the array at the end, not for each iteration. You'll use the output of the print statements for the written responses.

(**Hint:** You may want to add a house variable that is either True or False that can help you keep track of when each one owns a house or not.)

Technical details

You will not have to return anything for any of the functions except simulator(). You can pass numeric types to functions by value, but structs and array should be passed by pointer. This is very important!

You might need to define `printwealth()` and `printperson()` functions for debugging.

Input and output

The sample numbers for simulation have been provided for you and you should develop and test your simulation using those numbers. After the simulation is done though, you might want to explore different values of yearly earnings, mortgage interest rates, house prices and other parameters discussed above. Those values should be entered from file. Think about a good format for this input file. Additionally, the wealth arrays are best analyzed visually, so they need to be output in corresponding output files, one number per line.

Many people

After basic input and output are done, generalize your program to simulate lifetime wealth for several persons. Each person will have their own input file and output file associated with them.

Writing Response

Answer the following questions in writing:

1. What did you learn from this assignment that you didn't know before?
2. How much more debt did nfl end up paying compared to fl?
3. How many more years was nfl in debt compared to fl?
4. How much more money does fl have at the end of 40 years compared to nfl?
5. You don't need to test each function to answer this question, but which decision do you think had the greatest effect on the disparity between nfl and fl?
6. Come up with three life financial decisions (that were not mentioned in this assignment) that a non-financially literate person and a financially literate person might make differently. Specify which person would make which decision.
7. Plot the wealth of nfl and fl using your favorite graphing software (e.g. Excel).

Submit

- Main code file(s). Code should be clean, well-commented, have meaningful variable and function names, not have commented-out code. Code should be easily readable by a human; max number of nesting in ifs and loops is three. Main code should be broken into functions that perform logical parts of the job. Most importantly, the code should compile and run.

- A word file containing explanation of your program, known extensions and limitations, and instructions how to run the program, an example of your text input file.
- A .pdf file containing answers to written response part and a plot of wealth for two people under conditions described in the assignment.