In the struct savings, debt, debt paid, years in debt, savings interest, checking, downpayment, houseprice, loan, rent, debt interest, has house, years rented, debt additional payment, and house interest are defined. Has house will start at false and debt paid, years in debt, years rented, and loan will start at 0. All other members of the struct should be defined with a number other than zero in the main. Once the simulator is called on a person with a salary amount the simulator will open a file for writing and will write the first element of the array to the first line of the file. The simulator will enter a for loop that will run 40 times. A portion of the yearly salary will be added to the checking and savings accounts. Interest form savings account function will be called along with the debt function. If the hasHouse function is false and checking is less than downpayment, then rent function will be called. Otherwise, if checking is greater than downpayment then HasHouse will be set to true and the person’s loan will be set to the house price minus the downpayment. If HasHouse is true, then the house function will be called and loan amount and checking account will decrease. Wealth is calculated and set to the i element of the array. The createPerson function reads from a file and will take as input its text contents for the values of the struct and returns a person.

The program’s loan is diminished in less than 30 years and while the createPerson function runs, it doesn’t seem to read the values from the file correctly.

Example of input:

3000

.02

1000

10100

10

.0

25000

.15

650