Homework Week 1

September 7, 2021

1 Homework Week 1

1.1 Fibbonacci Maker

[1]:
```
def makeFib(n1,n2):

    numList = [n1,n2]

    for i in range(0,29):
        numList.append(numList[-1] + numList[-2])

    return numList

print(*makeFib(int(input("Input first number:")), int(input("Input second␣
 ↪number:"))))
```

```
Input first number: 10
Input second number: 12
```

```
10 12 22 34 56 90 146 236 382 618 1000 1618 2618 4236 6854 11090 17944 29034
46978 76012 122990 199002 321992 520994 842986 1363980 2206966 3570946 5777912
9348858 15126770
```

1.2 Mortgage Calculator: Monthly Payment and Graph

$

$$P = L\frac{c\,(1+c)^n}{(1+c)^n - 1} \tag{1}$$

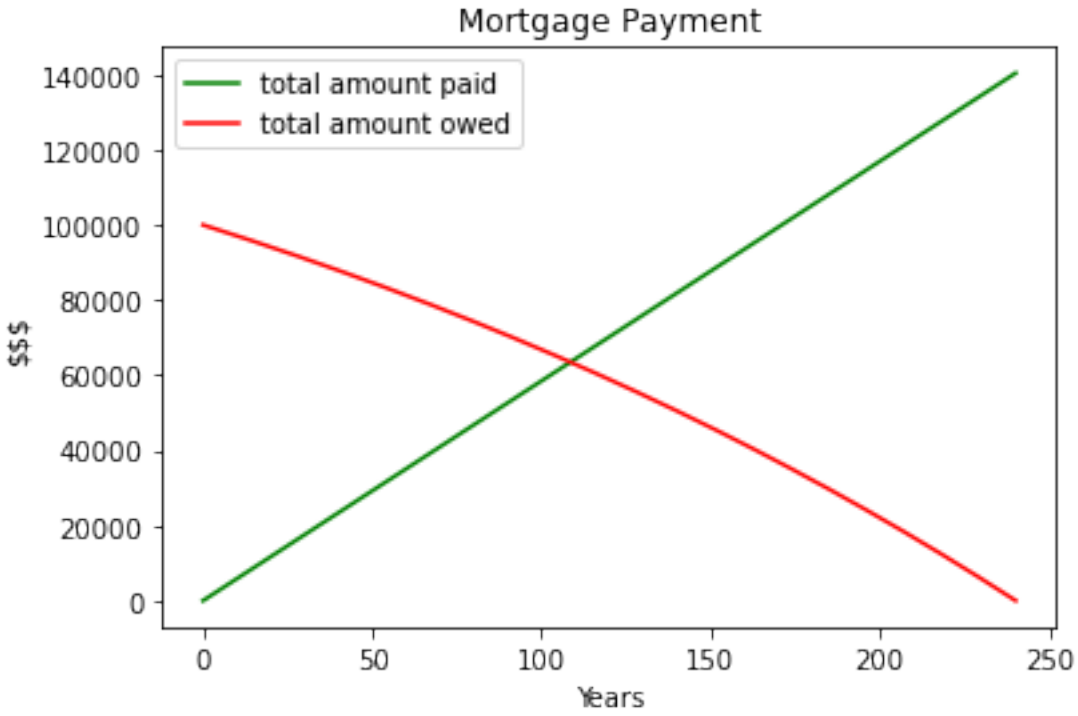
$

[2]:
```
import matplotlib.pyplot as plt

L = float(input("House Price ($)? "))
r = float(input("Mortgage rate (%)? "))/100
c = r/12
termLength = int(input("Term length (years)? ")) #total number of years spent␣
 ↪paying the mortgage, not the time between payments, which is assumed to be 1␣
 ↪month
n = 12 * termLength
```

```
P = L*(c*(1+c)**n)/((1+c)**n-1)

principleOwed = [L]
principlePaid = [0]

interestOwed = [0]
interestPaid = [0]

totalPaid = [0]
totalOwed = [L]

for i in range(n):

    newInterest = interestOwed[-1] #calculate new interest, starting with␣
 ↪previous amount owed

    interestAccrued = (principleOwed[-1] + interestOwed[-1])*c #calculate␣
 ↪accrued interest based on balance owed, interest owed, and interest rate
    newInterest += interestAccrued #add accrued interest

    interestPayment = P - L/n #pay off some of the interest
    interestPaid.append(interestPaid[-1] + interestPayment) #track interest paid

    newInterest -= interestPayment #apply interest payment
    interestOwed.append(newInterest) #track interest owed


    newBalance = principleOwed[-1] #calculate new balance, starting with␣
 ↪previous amount owed

    balancePayment = L/n #pay off some of the balance
    principlePaid.append(principlePaid[-1] + balancePayment) #track balance paid

    newBalance -= balancePayment #apply balance payment
    principleOwed.append(newBalance) #track balance owed

    totalPaid.append(principlePaid[-1] + interestPaid[-1]) #calculate total␣
 ↪amount paid
    totalOwed.append(principleOwed[-1] + interestOwed[-1]) #calculate total␣
 ↪amount owed

print("")
print("Monthly Payment:", "${:,.2f}".format(P))
#print("Total $$$ Paid to the Bank:", "${:,.2f}".format(totalPaid[-1]))
print("Total Interest Paid to the Bank:", "${:,.2f}".format(totalPaid[-1] - L))
```

```
plt.plot(range(len(principleOwed)), totalPaid, label = "total amount paid", c =␣
↪'g')
plt.plot(range(len(principlePaid)), totalOwed, label = "total amount owed", c =␣
↪'r')
plt.title('Mortgage Payment')
plt.ylabel('$$$')
plt.xlabel('Years')
plt.legend()
plt.show()
```

```
House Price ($)?  100000
Mortgage rate (%)?  3.6
Term length (years)?  20


Monthly Payment: $585.11
Total Interest Paid to the Bank: $40,426.75
```



1.3 Mortgage Calculator: House Value

$

$$L = P\frac{(1+c)^n - 1}{c\,(1+c)^n} \tag{2}$$

$

[3]:
```
P = float(input("Monthly Payment ($)? "))
r = float(input("Mortgage rate (%)? "))/100
c = r/12
termLength = int(input("Term length (years)? "))
n = 12 * termLength
L = P*((1+c)**n-1)/(c*(1+c)**n)

print("House Value:", "${:,.2f}".format(L))
print("Total Interest Paid:", "${:,.2f}".format(P*n - L))
```

```
Monthly Payment ($)?  585.11
Mortgage rate (%)?  3.6
Term length (years)?  20

House Value: $99,999.75
Total Interest Paid: $40,426.65
```