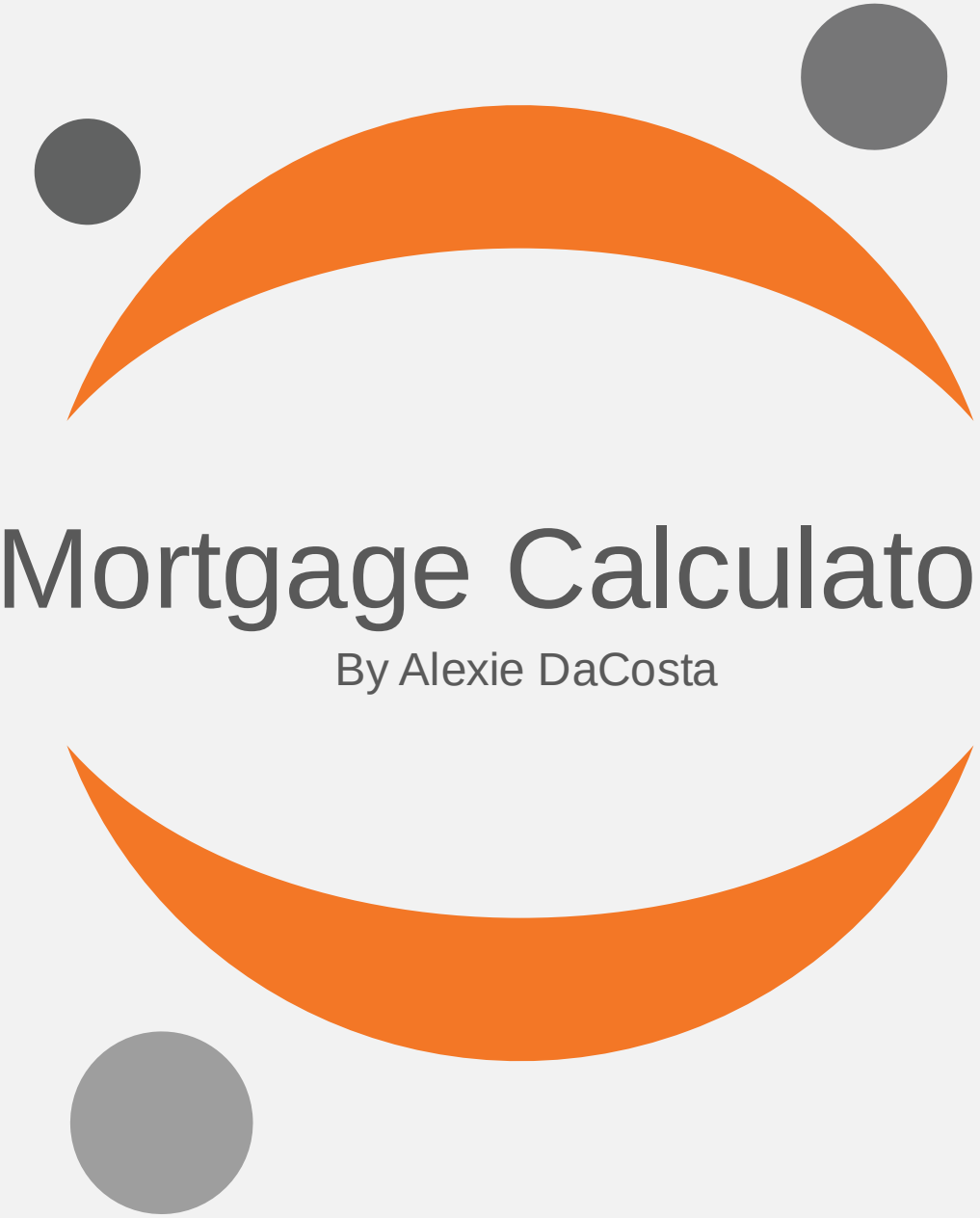


Mortgage Calculator

By Alexie DaCosta

Project Description

This code calculates the monthly, mortgage payments that the user can project to pay at a fixed rate. To execute this calculation, it first inquires the user's data, and then it inputs these values into the mortgage formula (See Fig. 1).

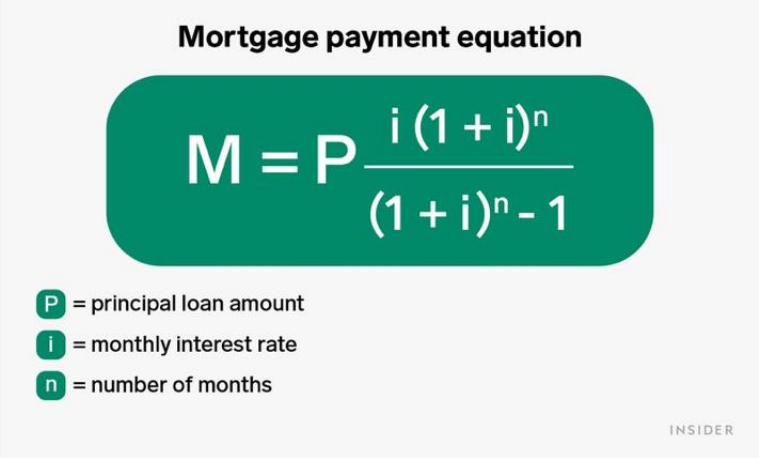


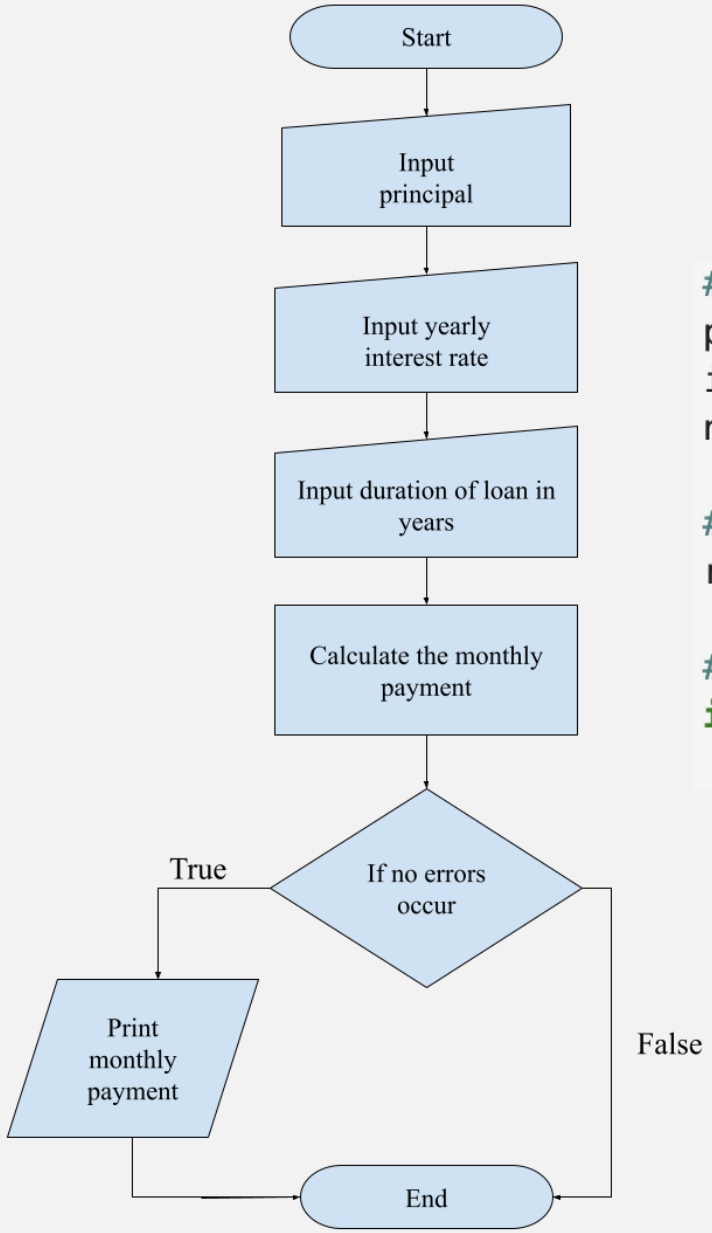
Fig 1.

Problem-Solution

Predatory mortgage lending—offered by creditors, brokers, and even home improvement contractors—takes advantage of borrowers' lack of understanding about loan terms (District of Columbia). Home buyers are given higher fees, interest rates, or penalties. It effectively hinders these borrowers' ability to repay debt. As a result, this practice leads to home buyers' losing their home in a foreclosure, benefiting the lender who gains control of the property as collateral. In a world of data manipulators who deliberately create inflated appraisals, quantitative reasoning is at the forefront of skills that mortgage loaners desire to have. This skill is the ability to perform mathematical calculations to make data-driven decisions in real-world scenarios (Bronx Community College). In fact, this mortgage calculator encapsulates that skillset. Loaners, using this calculator, can predict monthly payments, evaluate several different loans, and choose the best option that is available to them. This project equips loaners with a tool to make informed, financial decisions, helping to alleviate the problem of predatory mortgage lending.

Overview of the Algorithm

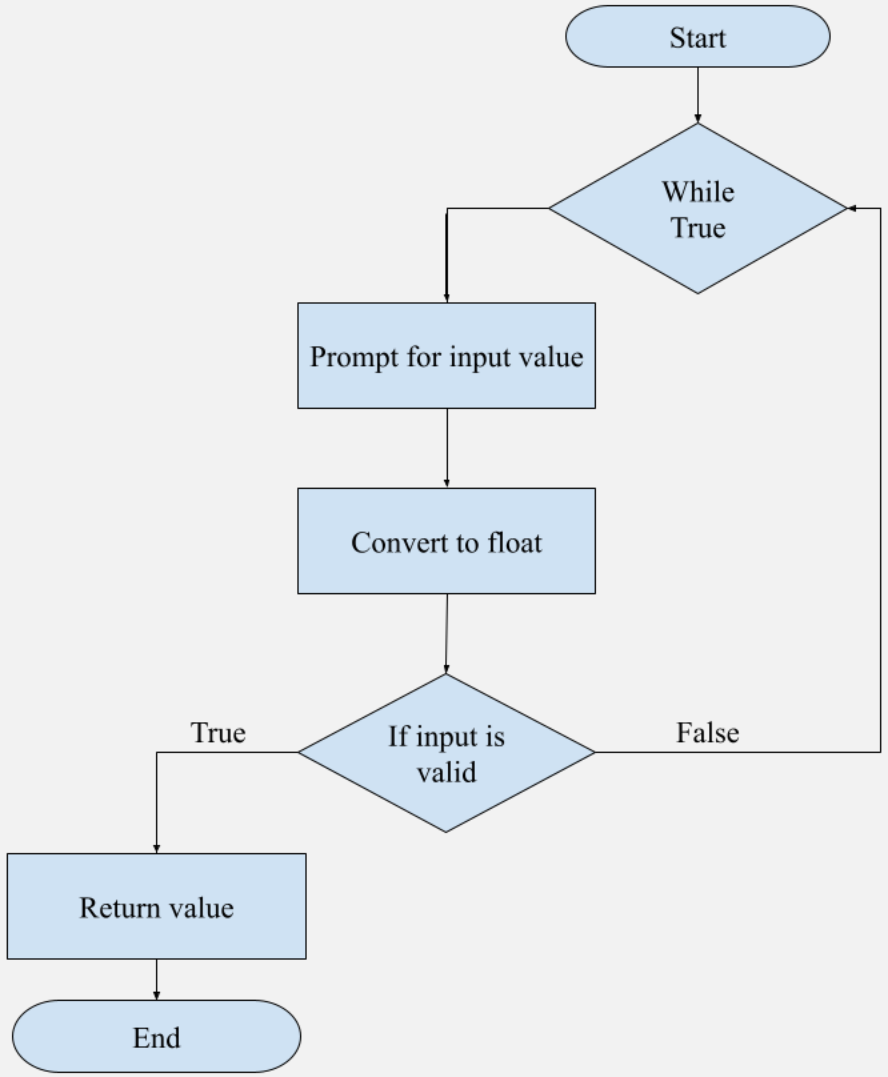
1. Main Program
2. Get Numerical Input Function
2a. Results
3. Mortgage Calculator Function
3a. Results



Algorithm: Main Program

```
#ask users for input / main
p = get_input_num("Enter the principal amount:")
i = get_input_num("Enter the yearly interest rate:")
n = get_input_num("Enter the length of the loan in years:")

# call the code and store the dictionary
result_dict = mortgage_calculator(p, i, n)

# the if statement checks that the other conditions did not fail
if result_dict["m"] > -1:
    print(f"Your monthly loan payment is ${result_dict["m"]}")
```

Algorithm:
Get Num Input Function




```
"""
reject alphabetical inputs / special characters, and accept numbers only
"""
def get_input_num(prompt):
    while True:
        try:
            val = float(input(prompt))
        except ValueError:
            print(f"Must enter a number, without special characters")
        else:
            break
    return val
```

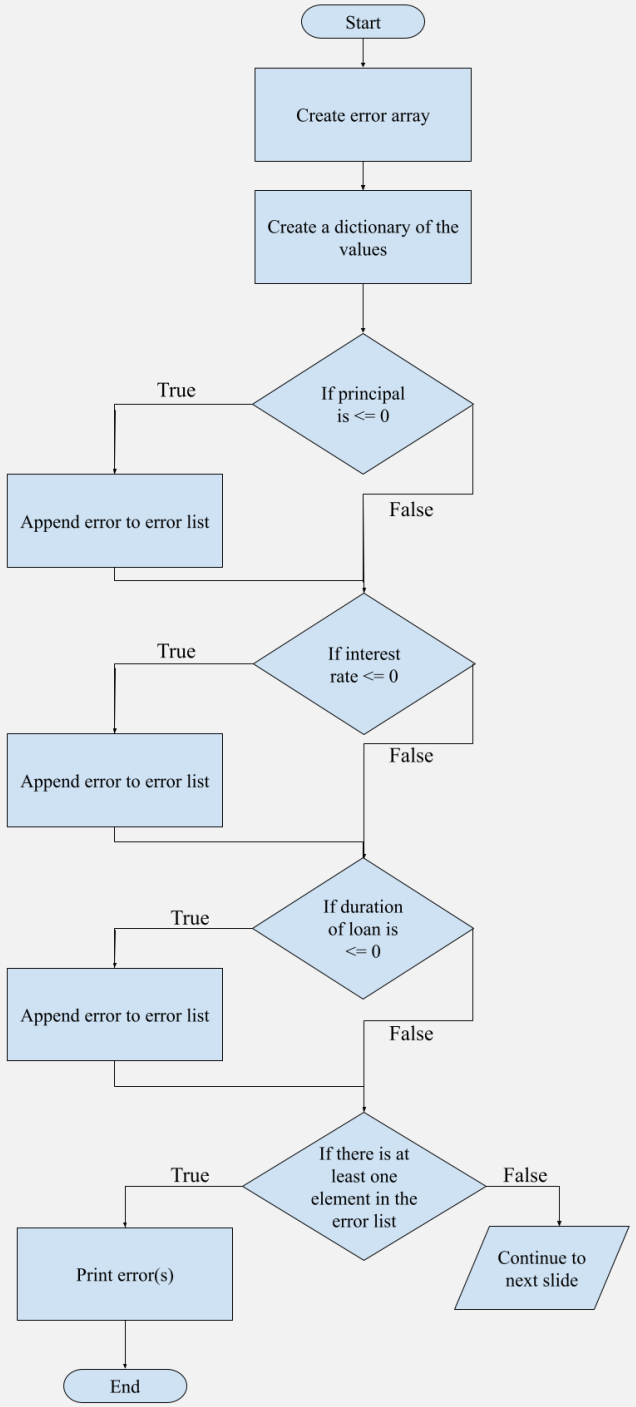
Results:
Get Num Input Function

Positive Case

```
Enter the principal amount: 425000
Enter the yearly interest rate: 6.95
Enter the length of the loan in years: 30
```

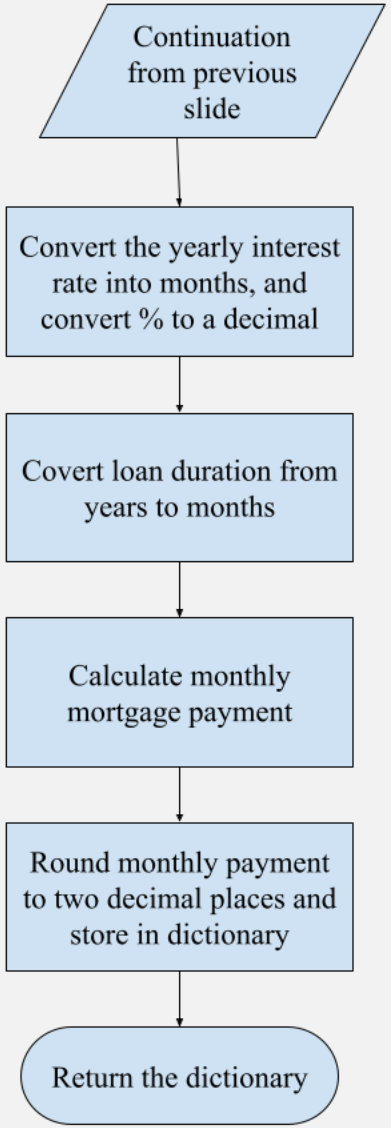
Negative Case

```
Enter the principal amount: Python
Must enter a number, without special characters
Enter the principal amount: $425,000
Must enter a number, without special characters
Enter the principal amount: 425000
Enter the yearly interest rate: six point nine five percent
Must enter a number, without special characters
Enter the yearly interest rate: 6.95
Enter the length of the loan in years: hi!
Must enter a number, without special characters
Enter the length of the loan in years: 30
```



Algorithm (1/2): Mortgage Calculator Function

```
def mortgage_calculator(p, i, n):
    error_list = []
    result_dict = {"p": p, "i": i, "n": n, "m":-1}

    # send error message to negative inputs and inputs = 0 to avoid an undefined answer
    if p <= 0:
        error_list.append("The principal loan amount must be greater than zero")

    if i <= 0:
        error_list.append("The interest rate must be higher than zero")

    if n <= 0:
        error_list.append("The duration of the loan must be longer than 0 years")

    if len(error_list) > 0:
        for error in error_list:
            print(error)

    # exit out of the method
        return result_dict
```



Algorithm (2/2): Mortgage Calculator Function

```
# convert yearly interest rate to monthly interest
# converting monthly interest rate into a decimal
mi = i / 100 / 12

# convert loan duration of years into months
n = n * 12

# calculate monthly payment
m = p * (mi * (1 + mi) ** n) / ((1 + mi) ** n - 1)

# round the decimal, m, to the hundreths place, representing cents
result_dict["m"] = round(m, 2)

# return the entire dictionary
return result_dict
```

Results:
Mortgage Calculator Function

Positive Case

```
Enter the principal amount: 425000
Enter the yearly interest rate: 6.95
Enter the length of the loan in years: 30
Your monthly loan payment is $2813.28
```

Negative Case

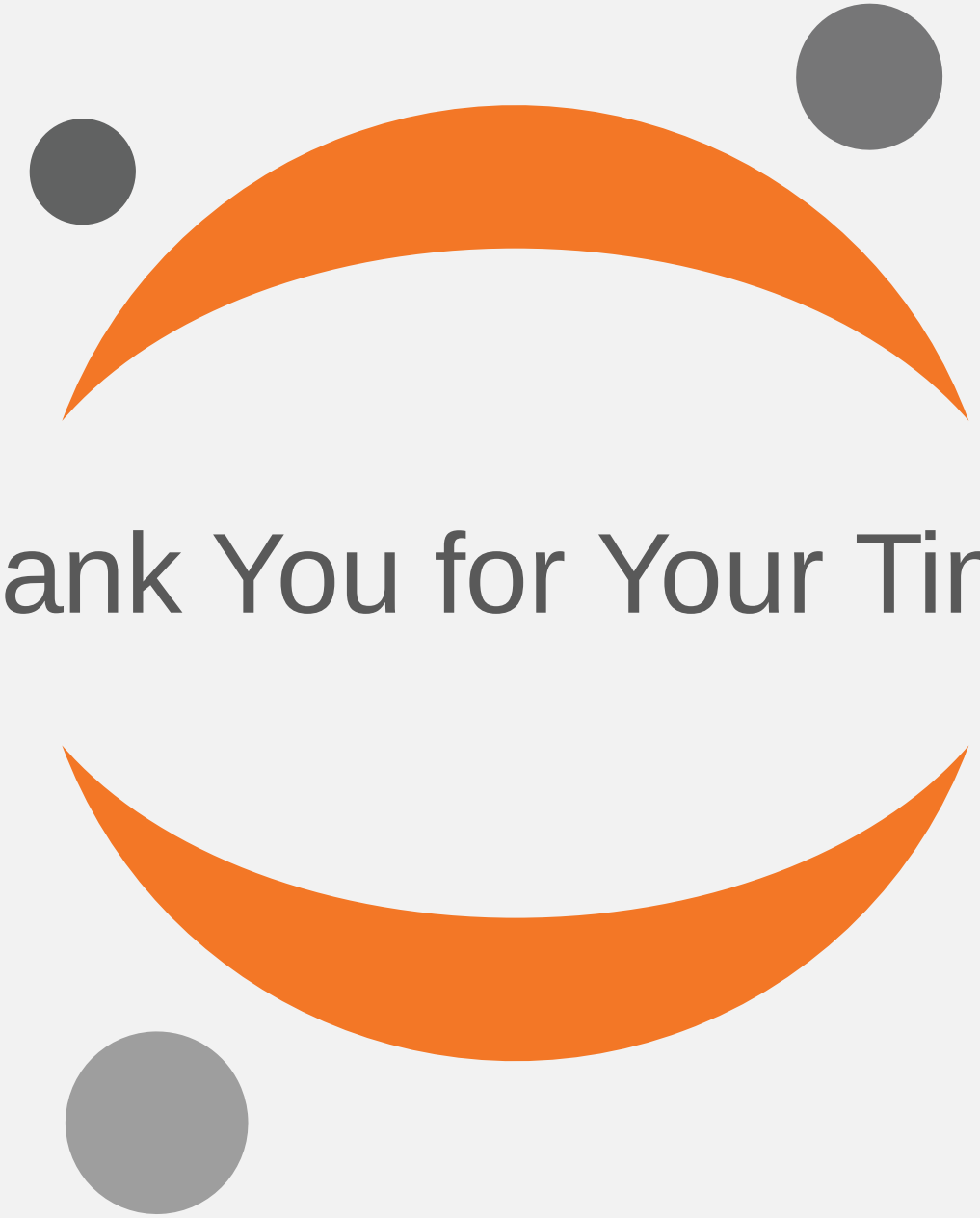
Variation 1

```
Enter the principal amount: 0
Enter the yearly interest rate: -3
Enter the length of the loan in years: -25
The principal loan amount must be greater than zero
The interest rate must be higher than zero
The duration of the loan must be longer than 0 years
```

Variation 2

```
Enter the principal amount: -25000
Enter the yearly interest rate: 6.95
Enter the length of the loan in years: 0
The principal loan amount must be greater than zero
The duration of the loan must be longer than 0 years
```


Q&A



Thank You for Your Time!

Works Cited

Grace, Molly. “Simple Mortgage Calculator: Estimate Your Monthly Payments.” *Business Insider*, Business Insider, 19 Nov. 2024, www.businessinsider.com/personal-finance/mortgages/mortgage-calculator.

“Predatory Mortgage Lending.” *District of Columbia*, oag.dc.gov/sites/default/files/2018-02/Predatory-Mortgage-Lending.pdf

“Quantitative Reasoning.” *Bronx Community College*, 31 July 2023, www.bcc.cuny.edu/academics/oie/general-education-assessment/quantitative-reasoning/.