MINISTRY OF EDUCATION, CULTURE AND RESEARCH OF REPUBLIC OF MOLDOVA

TECHNICAL UNIVERSITY OF MOLDOVA

FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS

DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

Project repor

Individual Work for DSA course

Loan Calculator

Profrsor: prof., Burlacu Natalia

Student: Diana Iachim, FAF-233

Chisinau, 2024

Content

Introduction

Financial literacy is crucial for making sound financial decisions. Unfortunately, many people lack the knowledge they need to understand loan terms, which can lead to financial difficulties down the road. Traditionally, loan documents have been filled with complex financial jargon and calculations, making it hard for the average person to grasp the true cost of a loan before they commit to it.

There is a clear need for a user-friendly tool to empower borrowers with knowledge. A loan calculator can address this need by providing financial transparency and informed decision making. This document will explore the problem of loan illiteracy, explain how loan calculators can address this problem, and delve into the details of how a loan calculator program works.

The first section, Problem Description, will discuss the challenges of loan illiteracy and the importance of financial transparency. The following section, Research Background, will explore the fundamental principles of borrowing money and the key factors to consider before taking on debt. It will also explain the different types of loans and the role of interest. Why Loans and Loan Calculators are Important will explain the various reasons why people borrow money and how loan calculators can help them make informed decisions. Why the Amortized Method is Preferred will discuss the two main methods for calculating loan payments and explain the advantages of the amortized method for both borrowers and lenders.

Algorithmic Breakdown of Amortized Loan Payments will provide a step-by-step explanation of the algorithm used in the loan calculator program, including the formulas and calculations involved. Solution Implementation will present and analyze the C code for the loan calculator, exploring its functionalities and limitations. Strengths, Weaknesses, and Room for Growth will discuss the program's strengths, weaknesses and potential areas for improvement. Flowchart will provide a visual representation of the program's logic.

This document aims to provide a comprehensive understanding of loan calculators and their role in promoting financial literacy. By empowering borrowers with knowledge, loan calculators can help them make informed decisions and avoid financial pitfalls.

Problem Description

Loan illiteracy hinders informed borrowing decisions.

People often lack the knowledge they need to make informed choices about loans. This can lead to financial difficulties down the road.

Traditionally, understanding loan terms has required wading through complex financial jargon and calculations. This makes it difficult for the average person to grasp the true cost of a loan before they commit.

There's a clear need for a *simple and accessible* tool to empower borrowers with knowledge.

Here's why a Loan Calculator is crucial:

- Financial Transparency: Many people don't fully grasp how factors like interest rates and loan terms affect their monthly payments and total loan cost. A loan calculator breaks down these complexities, allowing users to see the impact of different loan options clearly.

- Informed Decision Making: With a clear picture of potential monthly payments and total interest paid, borrowers can make informed decisions about whether a loan is truly affordable and aligns with their budget.

- Empowerment: A loan calculator equips borrowers with the knowledge they need to negotiate better loan terms and avoid predatory lending practices.

By providing a user-friendly way to understand loan options, a loan calculator can significantly improve financial literacy and empower people to make sound borrowing decisions.

Research Background

General Theory of Borrowing Money

This section will explore the fundamental principles of borrowing money and the key factors to consider before taking on debt.

1. The Need to Borrow:

There are various reasons why individuals might need to borrow money. These can be categorized into two main areas:

- Emergencies: Unexpected events like car repairs, medical bills, or sudden loss of income can necessitate borrowing to cover immediate financial needs.

- Planned Expenses: Sometimes, borrowing is a strategic financial decision to achieve long-term goals. This could include financing a higher education, purchasing a property, or making a significant investment.

2. Understanding Loan Types:

Borrowing money comes in various forms, each with its own characteristics and purposes. It's crucial to choose the loan type that best aligns with your specific needs and financial situation. Here's a breakdown of some common loan types:

- Personal Loans: Offered by banks, building societies, and finance companies, these are unsecured loans for various purposes. They typically have fixed interest rates and repayment terms.

- Secured Loans: These loans, like mortgages, are secured by an asset like a house. This means the lender can repossess the asset if repayments are not met. Secured loans often offer lower interest rates compared to unsecured loans.

- Student Loans: Designed to finance education, these loans have unique features like income-based repayments and interest accrual during the study period.

3. The Role of Interest:

Interest is the fee charged by the lender for borrowing their money. It's a crucial factor impacting the total cost of your loan. Here are some key points to remember:

- Interest Rates: The interest rate determines the percentage of the borrowed amount you pay in addition to the principal amount. Lower interest rates are generally more favorable.

- Fixed vs. Variable Rates: Fixed interest rates remain constant throughout the loan term, while variable rates can fluctuate based on market conditions.

4. Responsible Borrowing:

Borrowing money is a significant financial decision. Responsible borrowing practices require

careful consideration of the following:

- Repayment Ability: Ensure you have a realistic budget that can comfortably accommodate the monthly loan repayments.

- Impact on Credit Score: On-time loan repayments contribute positively to your credit score, whereas defaults can have a negative impact.

-Alternatives to Borrowing: Explore alternative solutions like savings plans or temporary adjustments to your budget before resorting to loans.

Why Loans and Loan Calculators are Important

Understanding loans, as i have mentioned in the previous section, is crucial for responsible financial management. People borrow money for various reasons, including:

1. Major purchases: Financing a car, house, or other significant investments can be facilitated through loans.

2. Emergencies: Unexpected events like medical bills or car repairs can necessitate borrowing to cover immediate needs.

3. Debt consolidation: Loans can help simplify managing multiple debts by consolidating them into a single payment.

4. Business ventures: Loans can provide capital to start or expand a business.

However, navigating the complexities of loans can be challenging. *Here's where loan calculators become valuable tools*:

- Informed Decisions: Loan calculators empower borrowers to compare loan options, understand the impact of interest rates and terms on repayments, and ultimately, choose the loan that best suits their financial situation.

- Transparency: By providing a clear breakdown of monthly payments, total interest paid, and overall loan costs, loan calculators promote financial transparency, allowing borrowers to make informed decisions before committing to a loan.

- Budgeting: Knowing the exact monthly payment amount helps borrowers budget effectively and ensures they can comfortably afford the loan repayments.

Why the Amortized Method is Preferred

There are *two main methods* for calculating loan payments: differentiated and amortized.

1. **Differentiated Method:** In this method, the borrower pays a fixed amount towards the principal each month, with the remaining payment covering the decreasing interest amount. While the total payment amount reduces over time, it can be initially high, making budgeting difficult.

2. **Amortized Method:** The amortized method is the more common and borrower-friendly approach. Here, the monthly payment remains constant throughout the loan term. This fixed payment includes both principal and interest. The portion allocated towards principal increases gradually as the loan progresses, while the interest portion decreases.

Advantages of the Amortized Method:

* Predictability: Fixed monthly payments make budgeting and financial planning easier for borrowers.

* Transparency: Loan calculators typically use the amortized method, providing borrowers with a clear picture of the total loan cost and the breakdown of principal and interest payments over time.

* Security: Knowing the exact amount due each month ensures borrowers avoid missed payments and potential late fees or penalties.

Benefits for Lenders:

While the amortized method offers advantages for borrowers, it also benefits lenders:

* Reduced Risk: Fixed payments make loan repayments more predictable for lenders, minimizing the risk of defaults.

* Efficiency: Amortized loans simplify loan servicing and management for lenders.

Algorithmic Breakdown of Amortized Loan Payments

In the following section we will delve deeper into the algorithm of the loan calculator, explaining each step in detail:

1. **Input Loan Parameters:**

+ **S** (Principal): This represents the total amount of money you borrow at the beginning of the loan. For example, if you borrow 200,000 rubles for a car.

+ **P** (Monthly Interest Rate): This is a crucial factor determining the cost of your loan. It's calculated by dividing the annual interest rate by the number of payments per year. For instance, if your annual interest rate is 0,12 , and your loan is structured for monthly payments (12 payments per year), then your monthly interest rate (P) would be 0,12 / 12 = 0,01 per month (converted into a decimal).

+ **N** (Total Number of Payments): This signifies the total duration of your loan in terms of payments. Continuing the car loan example with a 2-year term, if your loan is structured for monthly payments, then the total number of payments (N) would be 2 years * 12 payments/year = 24 months.

2. **Calculate Monthly Payment** (M):

This step employs the formula you provided:

$M = [S * (P(1+P)^N)]/[(1+P)^N - 1]$

Breaking Down the Formula: + **S**: This is the principal amount you borrowed (same as input).

+ **P**: This is the monthly interest rate you calculated earlier.

+ **N**: This is the total number of loan payments (same as input).

+ The entire expression in the bracket $P(1+P)^N$ calculates a factor that considers the interest rate and loan term.

+ Dividing by $(1+P)^N - 1$ normalizes the equation to determine the fixed monthly payment amount (M).

3. **Loop Through Each Payment Period (1 to N):**

Imagine iterating through each month of your loan term. Here's what happens in each iteration:

Initialize Variables:

+ Remaining Balance (B): This starts at the initial loan amount (S) you borrowed. In the car loan example, this would be 200,000 rubles.

+ Interest Payment (I): This is set to 0 initially, as you haven't paid any interest yet.

+ Principal Payment (Pr): Similar to interest, this also starts at 0 initially.

Calculate Interest Payment: $I = B * P$

This formula calculates the interest you owe for the current month.

+ **B** represents the remaining loan balance at the beginning of the month (which is initially the full principal).

+ **P** is the monthly interest rate you calculated earlier.

Calculate Principal Payment: $Pr = M - I$

This formula determines how much of your monthly payment actually goes towards reducing the loan amount (principal).

+ **M** is the fixed monthly payment you calculated in step 2.

+ **I** is the interest you just calculated for the current month.

Update Remaining Balance: $B = B - Pr$

This crucial step updates the remaining loan balance for the next month.

You subtract the principal payment (Pr) you just made from the remaining balance (B) at the beginning of the month.

Record Payment Details:

It's helpful to record details like:

* Month number

* Payment amount (M)

* Interest paid (I)

* Principal paid (Pr)

* Remaining balance (B) after the payment

This information can be compiled into an amortization schedule, which helps visualize how your loan progresses over time.

4. **End Loop**

Once the loop iterates through all payment periods (from 1 to N), you will have calculated the interest and principal portions of each monthly payment, along with the remaining loan balance at the end of each period.

Key Points to Remember:

+ The amortized method ensures a fixed monthly payment throughout the loan term.

+ The interest is calculated based on the remaining loan balance at the beginning of each month.

+ As you make payments, the principal portion increases, and the interest paid decreases due to the shrinking balance.

Solution implementation

Next I will present and analyse my solution for the problem stated above, using the algorithm described. This analysis delves deeper into the C code for the loan calculator, exploring its functionalities and limitations in more detail.

Note: GitHub repository will be available in the bibliography

Functionalities:

1. ***Menu Interface:***

* *displayMenu* function dynamically prints a menu with options for:

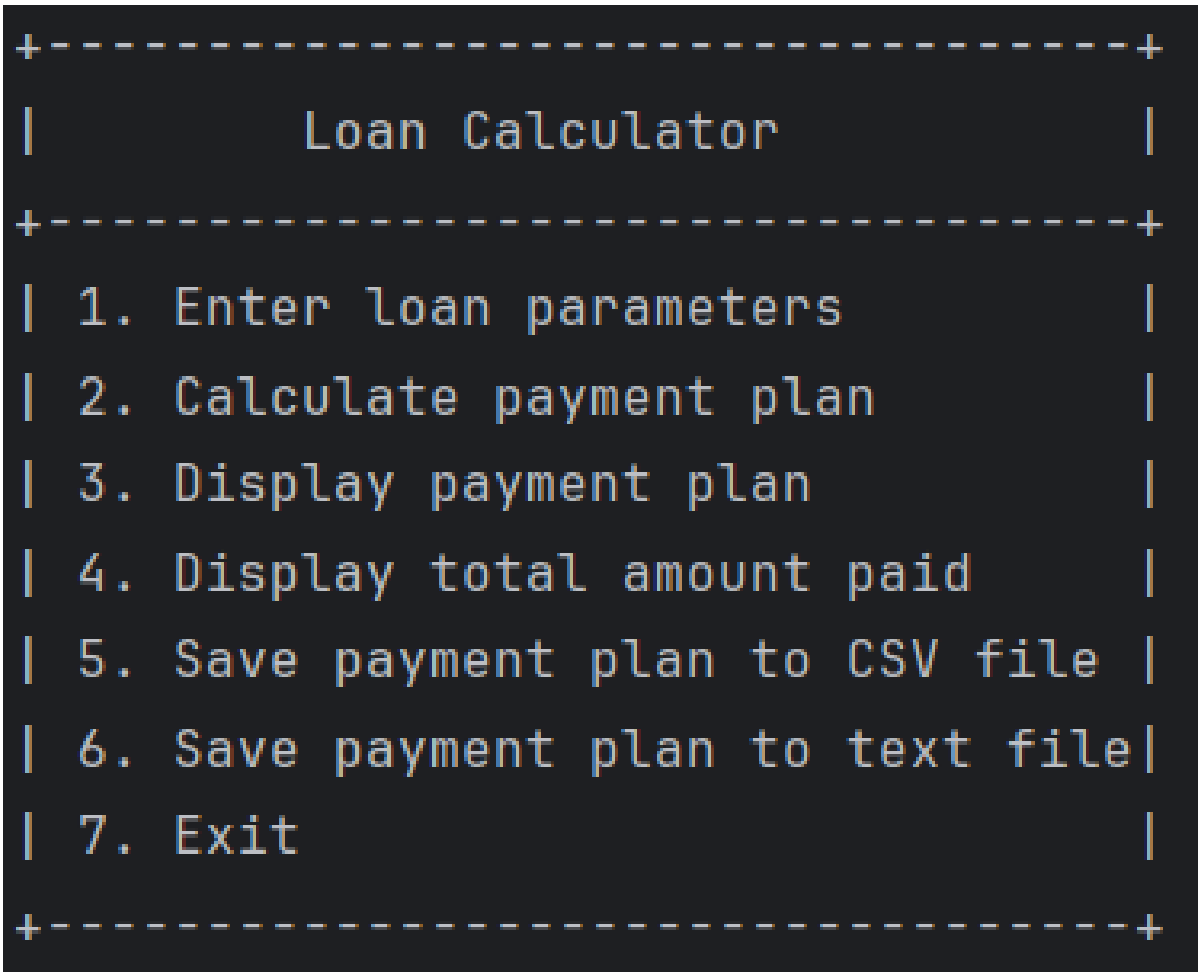

Figure 1 - Menu interface

* User input is obtained via *scanf* for choosing an option.

2. ***Loan Parameter Input:***

* Variables like *principal*, *inteRestrate*, *numPayments*, and *payment* are declared to store loan details and the calculated payment.

```
Enter the principal amount:12000
 Enter the annual interest rate (5 for 5%):7
 Enter the number of payments (120 for 10 years):
34
```

Figure 2 - Input example

* User input for principal, annual interest rate, and number of payments is prompted and stored in respective variables.

*The annual interest rate is converted to a monthly rate by dividing by 100 (percentage to decimal) and then by 12 (months in a year).

3. ***Payment Calculation:***

* The case 2 in the switch statement handles payment calculation.

* The loan payment formula

$payment = (principal * interestRate)/(1 - pow(1 + interestRate - numPayments))$

is used to calculate the monthly payment.

* This formula considers the principal amount, interest rate, and number of payments to determine the fixed monthly payment required to repay the loan over the term.

4. ***Payment Plan Display:***

* The case 3 in the switch statement handles displaying the payment plan.

* It checks if the payment has been calculated first (avoids displaying an empty plan).

* A table header with columns for Month, Payment, Interest, Principal, and Balance is printed.

* A loop iterates through each month (1 to numPayments).

+Interest paid for the month is calculated using $remainingBalance * interestRate$.

+Principal paid is calculated as $payment - interestPaid$.

+Remaining balance is updated by subtracting the principal paid.

+Details for each month (month number, payment, interest, principal, balance) are formatted and printed using *printf*.

+The total amount paid is accumulated throughout the loop.

5.***Total Amount Paid:***

* The *case 4* in the *switch* statement handles displaying the total amount paid.

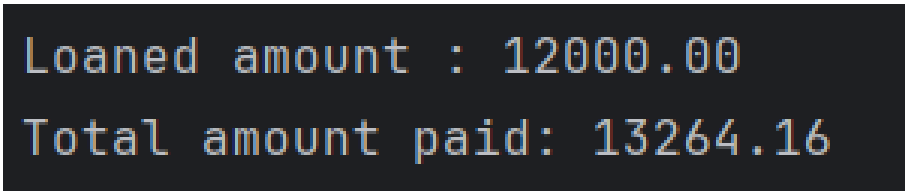

Figure 3 - Display total amount paid

* It checks if the payment has been calculated first.

* The total amount paid is calculated by multiplying the monthly payment by the number of payments (assuming all payments are made).

6. ***File Saving:***

* *Cases 5* (save to CSV) and *6* (save to text) handle saving the payment plan.

	A	B	C	D	E
1	Month	Payment	Interest	Principal	Balance
2	1	390.12	70	320.12	11679.88
3	2	390.12	68.13	321.99	11357.89
4	3	390.12	66.25	323.87	11034.02
5	4	390.12	64.37	325.76	10708.26
6	5	390.12	62.46	327.66	10380.6
7	6	390.12	60.55	329.57	10051.04
8	7	390.12	58.63	331.49	9719.54
9	8	390.12	56.7	333.43	9386.12
10	9	390.12	54.75	335.37	9050.75
11	10	390.12	52.8	337.33	8713.42
12	11	390.12	50.83	339.29	8374.13
13	12	390.12	48.85	341.27	8032.86
14	13	390.12	46.86	343.26	7689.59
15	14	390.12	44.86	345.27	7344.33
16	15	390.12	42.84	347.28	6997.04
17	16	390.12	40.82	349.31	6647.74
18	17	390.12	38.78	351.34	6296.39
19	18	390.12	36.73	353.39	5943

Figure 4 - Save payment plan to CSV file

*Both cases check if the payment has been calculated first.

*File pointers are used to open files named ”$loan_data.csv$” (CSV) and ”$loan_data.txt$” (text) for writing.

```
Payment Plan:
Month | Payment | Interest | Principal | Balance
1 | 390.12 | 70.00 | 320.12 | 11679.88
2 | 390.12 | 68.13 | 321.99 | 11357.89
3 | 390.12 | 66.25 | 323.87 | 11034.02
4 | 390.12 | 64.37 | 325.76 | 10708.26
5 | 390.12 | 62.46 | 327.66 | 10380.60
6 | 390.12 | 60.55 | 329.57 | 10051.04
7 | 390.12 | 58.63 | 331.49 | 9719.54
8 | 390.12 | 56.70 | 333.43 | 9386.12
9 | 390.12 | 54.75 | 335.37 | 9050.75
10 | 390.12 | 52.80 | 337.33 | 8713.42
11 | 390.12 | 50.83 | 339.29 | 8374.13
12 | 390.12 | 48.85 | 341.27 | 8032.86
13 | 390.12 | 46.86 | 343.26 | 7689.59
14 | 390.12 | 44.86 | 345.27 | 7344.33
15 | 390.12 | 42.84 | 347.28 | 6997.04
16 | 390.12 | 40.82 | 349.31 | 6647.74
17 | 390.12 | 38.78 | 351.34 | 6296.39
```

Figure 5 - Save payment plan to text file

*Error handling is included to check if the file opened successfully.

*The CSV format saves data separated by commas, while the text format displays a formatted table similar to the console output.

*Data for each month (month number, payment, interest, principal, balance) is written to the respective file.

*Files are closed after writing.

Limitations:

1. Error Handling:

+ The code lacks extensive error handling. It doesn't validate user input for:

- Invalid data types (e.g., entering strings instead of numbers)

- Negative values (e.g., negative principal amount)

+ Unexpected behavior might occur with invalid input.

2. Fixed File Names:

+The code uses fixed file names for saving the payment plan. It would be more flexible to allow users to specify their desired file names.

3. Limited Functionality:

+The program focuses on a single loan scenario. It doesn't offer functionalities like:

-Comparing different loan options with varying interest rates or terms.

-Calculating early payoff scenarios and potential savings.

Strengths, Weaknesses, and Room for Growth

The analyzed C code offers a user-friendly loan calculator program with a menu interface. While functional, there's significant potential for improvement. Let's delve into its advantages, limitations, and possibilities for expansion.

Strengths on Display:

* *Accessibility for All:* The program's clear menu interface makes it approachable for users with varying levels of financial knowledge. This allows them to navigate the program and access loan-related calculations with ease.

* *Accurate Calculations, Transparent Results:* The code leverages a well-established formula to calculate monthly loan payments. It then presents a detailed breakdown of the payment plan, clearly showing how each payment contributes to principal and interest. This transparency empowers users to understand their loan structure effectively.

* *Data on the Go:* The ability to save the payment plan in CSV or text format provides users with valuable data beyond the program itself. This allows them to store and analyze the information for record-keeping or sharing it with financial advisors.

Areas for Improvement:

* **Enhancing User Experience:** Currently, the code lacks robust error handling. This means invalid user input could lead to program crashes or unexpected behavior. Implementing data validation would significantly improve user experience by preventing such issues and ensuring smooth program operation.

* *Expanding Functionality:* The program currently focuses on a single loan scenario. Its functionality could be enriched by incorporating features like:

* Loan comparisons with varying interest rates and terms, allowing users to choose the most suitable option based on their financial goals.

* Early payoff calculations, empowering users to assess potential savings on interest by making additional payments.

* Integration with financial data for personalized calculations, creating a more tailored user experience.

* Support for different loan types (e.g., mortgage, auto) with their specific formulas.

* Visualization features like charts or graphs to represent the payment plan, providing users with a clearer visual understanding of their loan structure.

* Amortization schedule generation, offering a detailed breakdown of how the loan balance is reduced over time with each payment.

* **Maintaining and Improving:** Implementing data type validation for user input and incorporating error messages for unexpected situations would significantly enhance program robustness. Additionally, refactoring the code into well-defined functions would improve readability and maintainability. Separating functionalities like calculation, data display, and file saving would allow for easier updates and additions in the future. Unit testing individual functions would further ensure they work as intended, facilitating future modifications without introducing unintended bugs.

The Road to Expansion:

* *Loan Comparisons Made Easy:* By incorporating functionalities for comparing different loan options, the program could provide users with valuable insights and allow them to make informed financial choices.

* *Early Payoff Potential:* The ability to calculate potential savings on interest by making extra payments would be a valuable feature for users looking to pay off their loan faster.

* *Financial Data Integration:* Integrating with external financial data sources (e.g., bank accounts) could allow users to automatically populate loan parameters and generate personalized payment plans, streamlining the process.

* *A Multi-Loan Master:* Expanding the program to handle different loan types (mortgage, auto, etc.) with their specific calculations would create a more versatile tool.

* *Visualization for Clarity:* Including features for generating charts or graphs representing the payment plan would provide users with a more visual understanding of their loan structure.

By addressing the limitations and implementing these proposed expansions, the loan calculator code has the potential to become a comprehensive and valuable tool in various real-world applications, empowering users with financial clarity and informed decision-making.

Flowchart

Note : this will also be available in the GitHub repository

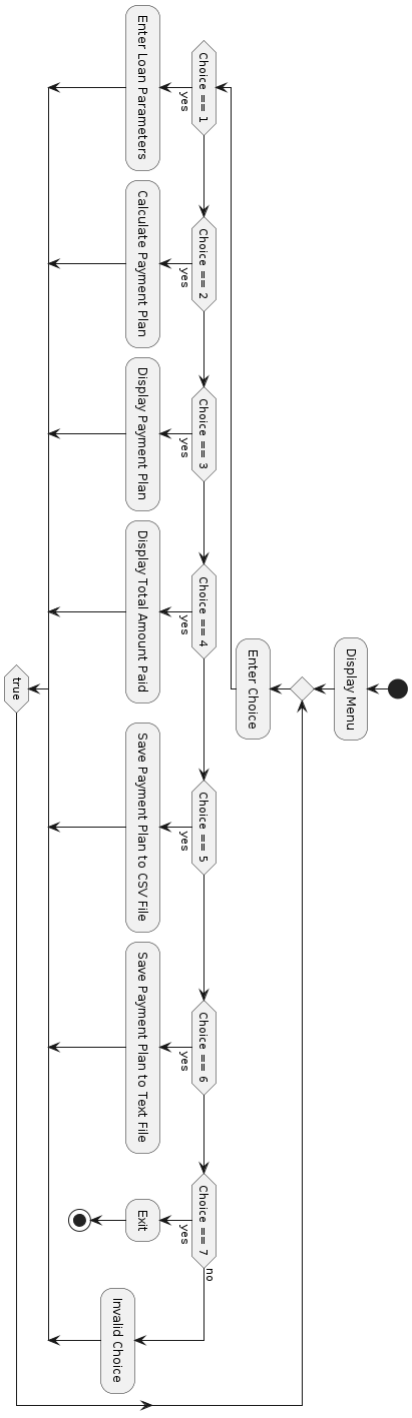

Figure 6 - Flowchart

Conclusions

This document has explored the importance of loan calculators in promoting financial literacy and empowering borrowers to make informed financial decisions. Loan illiteracy can lead to significant financial difficulties, highlighting the need for user-friendly tools that translate complex loan terms into clear and actionable information.

We examined the inner workings of a loan calculator program, delving into the amortized method for calculating loan payments and exploring the functionalities and limitations of a sample C code implementation. The analysis revealed the program's strengths in providing financial transparency and user-friendly features. However, there's also potential for improvement through enhanced error handling, expanded functionalities like loan comparisons and data visualization, and improved code maintainability.

By addressing these limitations and incorporating the proposed advancements, loan calculator programs can become even more comprehensive and valuable tools. Equipping borrowers with knowledge through financial education resources and user-friendly calculators can empower them to navigate the complexities of loans with greater confidence. This not only benefits individuals but also fosters a financially responsible society.

In conclusion, loan calculators play a vital role in promoting financial literacy and fostering informed borrowing practices. As these tools continue to evolve and integrate with new technologies, they have the potential to become even more powerful resources for borrowers, ultimately contributing to a more financially secure future.

Bibliography

[1] GitHub repository https://github.com/diana7376/Loan-Calculator.git.

[2] How to Calculate Loan Interest https://www.raiffeisen.ru/wiki/kak-rasschitat-procenty-po-kreditu/.

[3] Ultimate Guide to Your Credit Score and Credit Score Ranges https://www.nerdwallet.com/article/finance/credit-score-ranges-and-how-to-improve?ajs_uid=362cc18dd90ef0deaefbd6c7d8e2b7572a90e54f71f09e284c9f73e5865ad456&reg_gate_return=true.

[4] What Is a Loan, How Does It Work, Types, and Tips on Getting One https://www.investopedia.com/terms/l/loan.asp.

[5] Simple Loan Calculator https://www.creditkarma.com/calculators/personal-loans/simple-loan

[6] Amortization vs. Depreciation: What's the Difference? https://www.investopedia.com/ask/answers/06/amortizationvsdepreciation.asp

[7] Design a Loan Calculator https://www.geeksforgeeks.org/design-a-loan-calculator-using-javascript/