MS3111 Data Analytics with Excel VBA

Assignment Two

- **This assignment is not a team assignment. You are expected to complete the assignment by yourself. Discussions between classmates are encouraged, but you must not cross the line between discussion and plagiarism (such as sharing the answer in any way). Under no circumstances should you show your code to other students. All students involved in identified collaboration works will receive 0 marks for this assignment and may be subject to further disciplinary penalties (such as failing this course).**
- Save the workbook containing your work as *nnnnnnnn*.xlsm where *nnnnnnnn* is your CityU EID name (CityU email name). You are reminded to save the workbook as an Excel Macro-Enabled Workbook. Submit your Excel file via the designated link under the Assignments section in Canvas before 6:00 am on 18 March 2024. Only the last submitted file will be marked if you submit your file more than once. Late submissions will not be accepted.
- You must not change the appearance of Userform frmRepayment and the properties of any controls on the Userform at design time or runtime unless you are told explicitly to do so.
- You may enter your code into the included subs, but you must keep and not modify the code that is already contained in these subs.
- You must specify all utilised properties of Userform controls properly in your code. For example, the Value property of TextBox1 must be specified as TextBox1.Value. Missing the property specification of an object will be considered an error.
- Unless the assigned value of a variable is needed again by any subs while the application is still running, it must be declared inside the sub that uses the variable. If the value of a module-level variable is not to be reused by more than one sub, it will be considered an error.
- Redundant statements (including irrelevant statements or statements not associated with the listed tasks) will be considered errors.
- All code must be processed sequentially according to the order imposed by the task list. That is, Task 1ai must be executed before Task 1aii. Code not executed in the required task order will be considered an error.
- Marks will be deducted for the poor layout of the code. That includes non-properly named or declared variables, poor flow of the program, lack of comment statements, low readability, etc.

Tasks:

The formRepayment in workbook Assignment2_2023.xlsm is designed to compute and report the repayment information of car purchases. The user will specify the loan amount and terms at the application's runtime. All control objects have already been named. You must use these names in your codes. You are asked to provide code in the application for the following tasks:

Task 1: Before the formRepayment is drawn on screen at runtime:

a) Set the Frame control fmeMethodPay invisible programmatically.
b) Set the following properties of sbrDownPct programmatically:
   - Minimum value = 0
   - Large change value = 5
c) Populate lstMonthR with a list of interest rates from 0.0025 to 0.1500 with increments of 0.0025. All rates must be displayed in 4 decimal places, such as 0.0025, 0.0050, 0.0075, etc. You must accomplish this task using a relevant For loop or Do loop.
d) Set 0.01 as the current value of lstMonthR programmatically.

Task 2: When one or more items in lstModelPrice are selected at runtime, set fmeMethodPay visible. Otherwise, set fmeMethodPay back to invisible. {Note: the user may select or deselect a car model at runtime.}

Task 3: When an item in lstModelPrice is selected or deselected at runtime:

a) Determine the current total prices of the selected cars in lstModelPrice.
b) Assign the current total prices to a double-type module-level variable.
c) Uncheck both optDownPay and optFullPay programmatically.
d) Set sbrDownPct's value to 0 programmatically.
e) Set txtDownAmt's value to '$0' programmatically.

Task 4: When the value of sbrDownPct is changed at runtime:

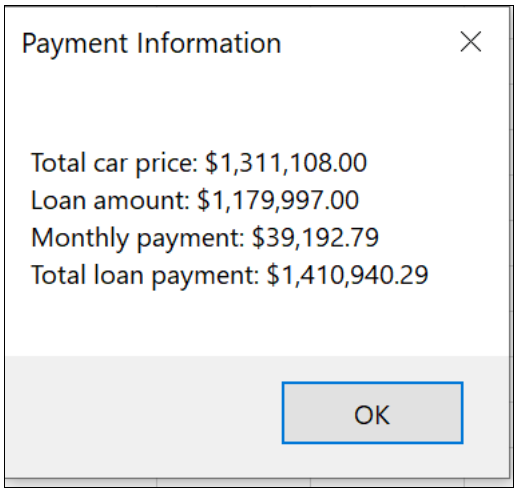
a) Display the current value of sbrDownPct in the caption of lblDownPct. The caption must end with a '%' symbol, such as '0%', '1%', etc.
b) Display the equivalent amount of the down payment in txtDownAmt as a roundup integer. The value shown must have a '$' symbol at the front and include a comma to separate thousands. For example, if the car price is $1,234,567. A 10% down payment shall be displayed in txtDownAmt as '$123,457'. If the down payment percentage is 0, show '$0' in txtDownAmt.

Task 5: When optFullPay is checked at runtime, programmatically set sbrDownPct's value to 100.

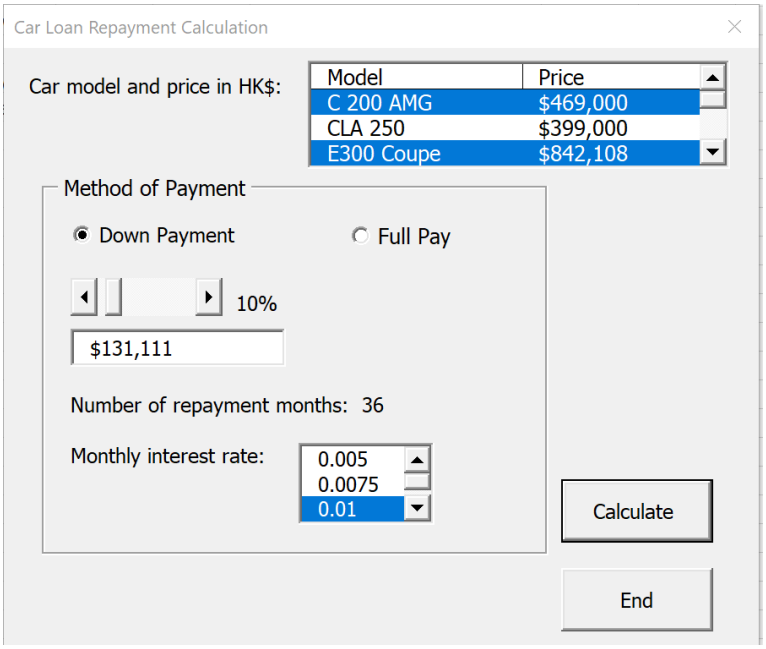
Task 6: When optDownPay is checked at runtime, programmatically sets sbrDownPct's value to 0.

Task 7: When btnCalculate is clicked:

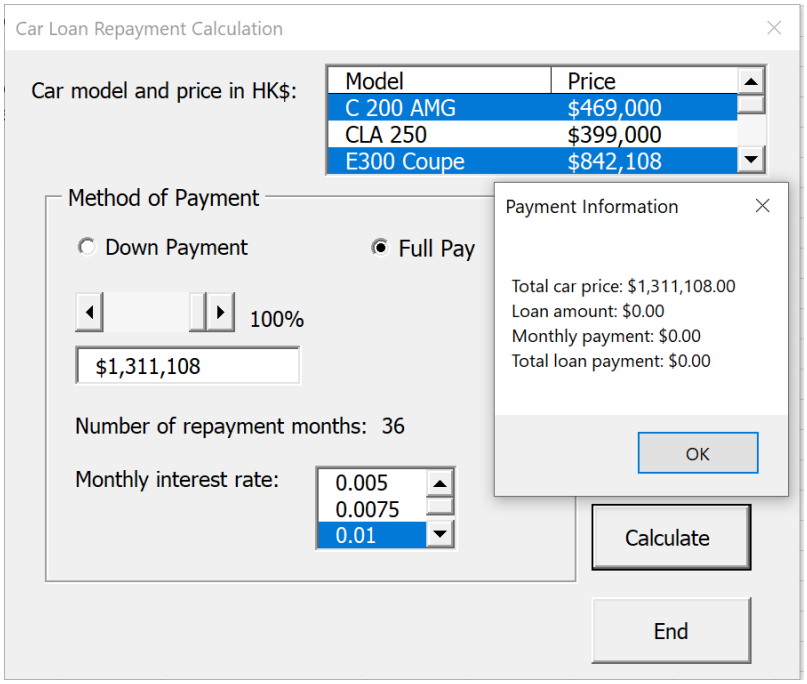
a) Compute the loan amount, which equals the current total car prices minus the down payment or full payment as selected.
b) Compute the monthly repayment using the VBA function PMT. {Note: VBA function PMT returns a negative value as payments are considered outgoing cash flows.}
c) Compute the total payment on the loan, which equals monthly repayment times the number of repayment months.
d) Display the above-computed values in a message dialog box like the one below. All money values must have a '$' symbol at the front and include a comma to separate thousands and in 2 decimal places:



The above are the results of the following settings:



Your code must display the correct values in the dialog box for a full-pay purchase:



-END-