1. Create a decision table to help the Municipal Bank decide whether or not to loan money to a customer. Include the criteria used by the bank to identify qualified applicants.
   * DISCLAIMER: I was unable to track down the criteria that the Municipal Bank utilizes to assess eligibility for a pedestrian loan. This assignment follows standard requirements for loan eligibility.

#	Requirements	Fulfilled	Unfulfilled	Unfulfilled	Unfulfilled	Unfulfilled	Unfulfilled	Unfulfilled	Unfulfilled
1	Credit Score <= 700?	True	False	True	True	True	True	True	True
2	Employment Duration >= 2 Years?	True	True	False	True	True	True	True	True
3	Debt-To-Income Ratio <36%?	True	True	True	False	True	True	True	True
4	Income Stable or Increasing?	True	True	True	True	False	True	True	True
5	Sufficient Collateral?	True	True	True	True	True	False	True	True
6	Loan Amount Within Limits?	True	True	True	True	True	True	False	True
7	Purpose of Loan Clearly Defined and Viable?	True	True	True	True	True	True	True	False
8	Passed?	True	False	False	False	False	False	False	False

* (Developed by Daryl) This does not feature every feasible combination of failed passes

2. Use pseudocode to write an algorithm for the decision table created in question one.

```
Class Loan {
  Constructor(creditScore, debtToIncomeRatio, employmentDuration,
incomeStability, sufficientCollateral, loanAmount, loanPurpose,
loanPurposeDefined) {
    This creditScore <- creditScore
    This debtToIncomeRatio <- debtToIncomeRatio
    This employmentDuration<- employmentDuration
    This incomeStability <- incomeStability
    This sufficientCollateral <- sufficientCollateral
    This loanAmount <- loanAmount
    This loanPurpose <- loanPurpose
    This loanPurposeDefined <- loanPurposeDefined
    This loaningLimits <- loaningLimits
  }

  Method assess() {
    Let creditScoreCriteria <- This creditScore >= 700
    Let debtToIncomeRatioCriteria <- This debtToIncomeRatio < 36
    Let employmentDuration <- This employmentDuration >= 2
    Let incomeStabilityCriteria <- This incomeStability == ‘consistent’ OR this
incomeStability == ‘increasing’
    Let collateralCriteria <- This sufficientCollateral >= This loanAmount
    Let loanPurposeCriteria <- This loanPurpose NOT == ‘’ AND This
loanPurposeDefined == true

    If(creditScoreCritera AND debtToIncomeRatioCriteria AND
employmentDuration AND incomeStabilityCriteria AND collateralCriteria AND
loanAmountCriteria AND loanPurposeCriteria) then.. {
```

```
        Return("We've decided to accept your loan request.")
      } Else then.. {
        Return("We've decided to decline your loan request.")
      }
    }
}

Let creditScore <- Convert to integer(Prompt("Enter your credit score: "))
Let debtToIncomeRatio <- Convert to integer(Prompt("Enter your debt-to-income
ratio (in a percentage): "))
Let employmentDuration <- Convert to integer(Prompt("Enter the amount of
years you have been employmed for: "))
Let incomeStability <- Convert to integer(Prompt("Enter your income stability
(increasing, consistent, decreasing): "))
Let sufficientCollateral <- Convert to integer(Prompt("Enter the value of your
collateral: "))
Let loanAmount <- Convert to integer(Prompt("Enter the loan amount: "))
Let loanPurpose <- Convert to integer(Prompt("Enter the loan purpose: "))
Let loan
Let loaningLimits <- Convert to boolean(Prompt("(To banker) Is the loan amount
within limits? (true/false): "))
Let loanPurposeDefined <- Convert to boolean(Prompt("(To banker) Is the loan
clearly defined and viable? (true/false): "))

Let loanProspect <- new Loan(creditScore, debtToIncomeRatio,
employmentDuration, incomeStability, sufficientCollateral, loanAmount,
loanPurpose, loaningLimits, loanPurposeDefined)

Let choice <- loanProspect.assess()

Print(choice)
```