Mainline logic

1. Initialize the loop control variable before entering the loop.
2. Test the value of the loop control variable, and if the condition is met, proceed into the body of the loop.
3. Modify the loop control variable within the loop to eventually satisfy the condition that causes the loop to end.

There is **total 3 problems**, please give your answers to the below questions

Problem #1:

```
// This pseudocode is intended to determine whether students have
// passed or failed a course; student needs to average 60 or
// more on two tests.
start
   Declarations
      num firstTest
      num secondTest
      num average
      num PASSING = 60
   output "Enter first score or 0 to quit "
   while firstTest not equal to 0
      output "Enter second score "
      input secondTest
      average = firstTest + secondTest / 2
      ouput "Average is ", average
      if average >= PASSING then
         output "Pass"
      else
         output "Fail"
      endif
      output "Enter first score or 0 to quit "
   endwhile
stop
```

1. Which variable controls the loop? ***firstTest variable***
2. Is the loop control variable properly initialized before entering the loop? If yes, what is the command to initiate it? If no, how you initialize it?

It is not; you have the initialize the loop by gaining user input. In the problem entered above there is no way for the user to enter data to be stored as 'firstTest' variable.

3. Is the variable that controls the loop correctly tested?

You cannot properly test it without being able to enter the loop.

4. Within the main loop, is there an action that updates the control variable to determine whether the loop continues or exits? If yes, what is the command? If no, how you update it?

Yes, the 'endif' output determines whether we return to the beginning of the program, or end by selecting '0 to quit'.

5. Besides errors in step 1-4, are there any other errors in this program? If yes, what are they? What are your actions to correct them?

```
// This pseudocode is intended to determine whether students have
// passed or failed a course; student needs to average 60 or
// more on two tests.
start
   Declarations
      num firstTest
      num secondTest
      num average
      // Changing num to constant to be sure nobody can change the
      // passing allowance
      constant PASSING = 60
   output "Enter first score or 0 to quit "
   // you need to call for user input to be able to be to
   // enter/control the loop
   input firstTest
   // changed the not equal to words to simply being the symbol shorthand
   while firstTest <> 0
      output "Enter second score "
      input secondTest
      // PEMDAS is used in all mathematics
      average = (firstTest + secondTest) / 2
      // the 't' was missing from the output
      output "Average is ", average
      if average >= PASSING then
         output "Pass"
      else
         output "Fail"
      endif
      output "Enter first score or 0 to quit "
   endwhile
stop
```

Problem #2:

```
// This pseudocode is intended to display employee net pay values.
// All employees have a standard $45 deduction from their checks.
// If an employee does not earn enough to cover the deduction
// an error message is displayed.
start
   Declarations
      string name
      num hours
      num rate
      string DEDUCTION = 45
      string EOFNAME = "ZZZ"
      num gross
      num net
   output "Enter first name or ", EOFNAME, " to quit"
   input name
   if name not equal to EOFNAME
      output "Enter hours worked for ", name
      input hours
      output "Enter hourly rate for ", name
      input rate
      gross = hours * rate
      net = gross - DEDUCTION
      while net > 0 then
         output "Net pay for ", name, " is ", net
      else
          output "Deductions not covered. Net is 0."
      endwhile
      output "Enter next name or ", EOFNAME, " to quit"
      input name
   endif
   output "End of job"
stop
```

1. Which variable controls the loop?

The name variable controls the loop.

2. Is the loop control variable properly initialized before entering the loop? If yes, what is the command to initiate it? If no, how you initialize it?

No, because in the original code the 'while loop' was written as an 'if' statement instead.

3. Is the variable that controls the loop correctly tested?

No, while it is initialized properly with its variable, the loop cannot repeat input due to it missing the 'while' at the main loop.

4. Within the main loop, is there an action that updates the control variable to determine whether the loop continues or exits? If yes, what is the command? If no, how you update it?

We cannot consider it to be an actual loop because the 'if-else" is replaced with a while, it has a block to be able to update the loop within.

5. Besides errors in step 1-4, are there any other errors in this program? If yes, what are they? What are your actions to correct them?

```
start
   Declarations
        string name
        num hours
        num rate
        // deduction should either be a num or a const
        // i have changed it to a const because it should
        // stay constant
        constant DEDUCTION = 45
        string EOFNAME = "ZZZ"
        num gross
        num net
   output ("Enter first name or ", EOFNAME, " to quit")
   input name
   // changed 'not equal to' to make it the shorthand
   // as well as changed the if to while so that we have
   // the ability to enter the loop through our name variable
   while name <> EOFNAME
        output "Enter hours worked for ", name
        input hours
        output "Enter hourly rate for ", name
        input rate
        gross = hours * rate
        net = gross - DEDUCTION
        // changed while to if to be a true 'if-else' statement
        if net >= 0 then
           output ("Net pay for ", name, " is ", net)
        else
           output ("Deductions not covered. Net is 0.")
        endwhile
        output ("Enter next name or ", EOFNAME, "to quit")
        input name
   endif
   output "End of job"
stop
```

Problem #3:

```
// This pseudocode segment is intended to compute and display
// the cost of home ownership for any number of users
// The program ends when a user enters 0 for the mortgage payment
start
   Declarations
      num mortgagePayment
      num utilities
      num taxes
      num upkeep
      num total
   startup()
   while mortgagePayment not equal to 0
      MainLoop()
   endwhile
   finishUp()
stop

startUp()
   output "Enter your mortgage payment or 0 to quit"
   input mtgPayment
return

mainLoop()
   output "Enter utilities"
   input utilities
   output "Enter taxes"
   input taxes
   output "Enter amount for upkeep"
   input upkeep
   total = mortgagePayment + utilities + taxes + upkeep
   output "Total is ", total
return

finishUp()
   output "End of program"
return
```

1. Which variable controls the loop? ***mortgagePayment created within the startup() module***
2. Is the loop control variable properly initialized before entering the loop? If yes, what is the command to initiate it? If no, how you initialize it?

No, because of the incorrect spelling of the variable "mortgagePayment" within the startup() module.

3. Is the variable that controls the loop correctly tested?

It would be considered correctly tested if mortgage payment had received the proper input rather than 'mtgPayment' receiving the input.

4. Within the main loop, is there an action that updates the control variable to determine whether the loop continues or exits? If yes, what is the command? If no, how you update it?

There is no way to see if the loop continues or exits due to not containing a end program statement.

5. Besides errors in step 1-4, are there any other errors in this program? If yes, what are they? What are your actions to correct them?

```
start
   Declarations
        num mortgagePayment
        num utilities
        num taxes
        num upkeep
        num total
   startup()
   // shorthanded 'not equal to'
   while mortgagePayment <> 0
        // changed the 'm' to lowercase
        mainLoop()
   endwhile
   finishUp()
stop

startup(mortgagePayment)
   output "Enter your mortgage payment or 0 to quit"
   // changed 'mtgPayment' to 'mortagePayment'
   input mortagePayment
return

mainLoop(mortgagePayment, utilities, taxes, upkeep, total)
   output "Enter utilities"
   input utilities
   output "Enter taxes"
   input taxes
   output "Enter amount for upkeep"
   input upkeep
   total = mortgagePayment + utilities + taxes + upkeep
   output "Total is ", total
return

finishUp()
   output "End of program"
return
```