MODULE NAME:	**MODULE CODE:**
INTRODUCTION TO PROGRAMMING LOGIC	IPRG5111

STUDENT NAME:	**STUDENT NUMBER**
LESEGO SEBAKO	ST10493865

QUESTION 1

```
START

    // Declartion
    NUM totalSeconds
    NUM wholeHours
    NUM wholeMinutes
    NUM remainingSeconds
    NUM partialHours
    NUM partialMinutes
    NUM tempMinutes

    // Input
    OUTPUT "Enter the total number of seconds"
    INPUT totalSeconds

    // Calculate hours
    partialHours = totalSeconds / 3600
    wholeHours = partialHours DIV 1
    partialHours = partialHours MOD 1

    // Convert partial hours to minutes
    tempMinutes = partialHours * 60
    wholeMinutes = tempMinutes DIV 1
    partialMinutes = tempMinutes MOD 1

    // Convert partial minutes to seconds
    remainingSeconds = partialMinutes * 60

    // Output
    OUTPUT "Equivalent time is:"
    OUTPUT wholeHours, "hour(s),", wholeMinutes, "minute(s),", remainingSeconds, "second(s)"

STOP
```

QUESTION 2

Q - 2.1

```
START

   // Declarations
   NUM originalNum, digit1, digit2, digit3, digit4, digit5
   BOOLEAN isPalindrome

   // Input
   OUTPUT "Enter a 5-digit number:"
   INPUT originalNum

   // Calculation: Extract each digit
   digit1 = originalNum DIV 10000
   remaining = originalNum MOD 10000

   digit2 = remaining DIV 1000
   remaining = remaining MOD 1000

   digit3 = remaining DIV 100
   remaining = remaining MOD 100

   digit4 = remaining DIV 10
   digit5 = remaining MOD 10

   // Determine palindrome
   isPalindrome = (digit1 == digit5) AND (digit2 == digit4)

   // Boolean output
   OUTPUT "Is the number a palindrome? "

   IF isPalindrome THEN
      OUTPUT "TRUE"
   ELSE
      OUTPUT "FALSE"
   END IF

STOP
```

QUESTION 2

Q - 2.2

```
START

    // Declaration
    STRING customerName, customerType, cardNumber
    NUM purchaseAmount, discountPercent, discountAmount, taxAmount, subTotal, finalAmount

    //Declaration - Input & Output
    OUTPUT "ROOTED"
    OUTPUT "Enter customer name:"
    INPUT customerName
    OUTPUT "Enter customer type (Bulk/Regular):"
    INPUT customerType
    OUTPUT "Enter purchase amount (R):"
    INPUT purchaseAmount

    // Get Card Number based on type
    IF customerType == "Bulk" THEN
        OUTPUT "Enter membership number:"
        INPUT cardNumber
    ELSE IF customerType == "Regular" THEN
        OUTPUT "Enter loyalty card number:"
        INPUT cardNumber
    END IF

    // Bulk Customer
    IF customerType == "Bulk" THEN
        IF purchaseAmount < 6000 THEN
            discountPercent = 6
        ELSE IF purchaseAmount < 12000 THEN
            discountPercent = 15
        ELSE
            discountPercent = 30
        END IF
        taxAmount = 0

    //Regular Customer
    ELSE IF customerType == "Regular" THEN
        IF purchaseAmount < 1500 THEN
            discountPercent = 4
        ELSE IF purchaseAmount < 3000 THEN
            discountPercent = 6
        ELSE
            discountPercent = 8
        END IF
    END IF

    // Calculations
    discountAmount = purchaseAmount * (discountPercent / 100)
    subTotal = purchaseAmount - discountAmount
    IF customerType == "Regular" THEN
        taxAmount = subTotal * 0.07
    ELSE
        taxAmount = 0
    END IF
    finalAmount = subTotal + taxAmount
    // Print Receipt
    OUTPUT "ROOTED"
    OUTPUT "--------------------------"
    OUTPUT "Customer Name: ", customerName
    OUTPUT "Customer Type: ", customerType
    OUTPUT "Purchase Amount: R ", purchaseAmount
    IF customerType == "Regular" THEN
        OUTPUT "Loyalty Card Number: ", cardNumber
    ELSE
        OUTPUT "Membership Number: ", cardNumber
    END IF

    OUTPUT ""
    OUTPUT "Discount Applied: ", discountPercent, "%"
    OUTPUT "Discount Amount: R ", discountAmount
    IF customerType == "Regular" THEN
        OUTPUT "Tax Applied: 7%"
        OUTPUT "Tax Amount: R ", taxAmount
    END IF

    OUTPUT ""
    OUTPUT "Sub-Total: R ", subTotal
    OUTPUT "Final Amount to be Paid: R ", finalAmount
    OUTPUT "(without Tax)"  // Only shown for Bulk customers
    OUTPUT "--------------------------"

STOP
```

QUESTION 3   Q - 3.1

```
START

    // Declarations
    NUM birthYear, currentYear, year
    BOOLEAN isLeapYear

    // Initialize current year
    currentYear = 2025  // Current year

    // Input
    OUTPUT "Enter your year of birth:"
    INPUT birthYear

    // Output leap years header
    OUTPUT "Leap years from ", birthYear, " to ", currentYear, ":"

    // Loop through years
    FOR year = birthYear TO currentYear

        // Determine leap year
        IF (year MOD 4 == 0 AND year MOD 100 != 0) OR (year MOD 400 == 0) THEN
            isLeapYear = TRUE
        ELSE
            isLeapYear = FALSE
        END IF
        // Output leap years
        IF isLeapYear THEN
            OUTPUT year
        END IF
    END FOR

STOP
```

QUESTION 3

Q - 3.2

```
START

    // Declarations
    NUM secretNumber, userGuess, guessCount
    STRING playAgain
    BOOLEAN gameActive

    // Main game loop
    DO
        // Initialize game
        secretNumber = RANDOM(1, 10)
        guessCount = 0
        gameActive = TRUE
        OUTPUT "Welcome to the Mystery Number Challenge!"
        OUTPUT "I've selected a number between 1 and 10. Can you guess it?"

        // Single game loop
        WHILE gameActive == TRUE
            // Get user guess
            OUTPUT "Enter your guess (1-10, or -1 to quit):"
            INPUT userGuess
            guessCount = guessCount + 1

            // Check for exit command
            IF userGuess == -1 THEN
                OUTPUT "Game exited."
                gameActive = FALSE

            // Check guess
            ELSE IF userGuess == secretNumber THEN
                OUTPUT "Congratulations! You guessed the correct number!"
                OUTPUT "Total guesses: ", guessCount
                gameActive = FALSE

            ELSE IF userGuess < secretNumber THEN
                OUTPUT "Too low! Try again."

            ELSE IF userGuess > secretNumber THEN
                OUTPUT "Too high! Try again."
            END IF
        END WHILE

        // Play again prompt
        OUTPUT "Would you like to play again? (yes/no)"
        INPUT playAgain
    WHILE playAgain == "yes"

    OUTPUT "Thanks for playing!"

STOP
```