START MathsTest Program

CLASS MathsTest

  METHOD __init__

    SET var_score TO 0

    SET var_results TO empty list

    SET var_correct_count TO 0

    SET var_questions TO 0

    SET var_max_num TO 0

  END METHOD

  METHOD choose_difficulty

    LOOP forever

      DISPLAY "Select a difficulty:"

      DISPLAY "1) Easy  2) Medium  3) Hard"

      INPUT choice

      CONVERT choice TO lowercase

      IF choice is "1" OR "easy" OR "e" THEN

        SET var_questions TO 5

        SET var_max_num TO 10

        DISPLAY "Easy mode selected!"

        BREAK LOOP

      ELSE IF choice is "2" OR "medium" OR "m" THEN

        SET var_questions TO 10

        SET var_max_num TO 20

```
        DISPLAY "Medium mode selected!"

        BREAK LOOP

    ELSE IF choice is "3" OR "hard" OR "h" THEN

        SET var_questions TO 15

        SET var_max_num TO 50

        DISPLAY "Hard mode selected!"

        BREAK LOOP

    ELSE

        DISPLAY "Invalid choice! Enter 1, 2 or 3."

    END IF

  END LOOP

END METHOD


METHOD create_question(min_num, max_num)

  GENERATE random num1 BETWEEN min_num AND max_num

  GENERATE random num2 BETWEEN min_num AND max_num

  RANDOMLY CHOOSE operator FROM ['+', '-']


  IF operator IS '-' AND num1 < num2 THEN

    SWAP num1 AND num2

  END IF


  RETURN STRING "{num1} {operator} {num2}"

END METHOD


METHOD ask_question(question)
```

```
        RECORD start time

        INPUT answer FROM user ("What is {question}?")

        RECORD end time

        CALCULATE time_taken AS integer of (end - start)

        CALCULATE correct_answer BY evaluating question


        RETURN (answer EQUALS correct_answer), time_taken
    END METHOD


    METHOD play
        DISPLAY "Welcome to Greg's Maths Test!"

        CALL choose_difficulty()


        FOR q FROM 1 TO var_questions
            DISPLAY current score

            DISPLAY "Question q of var_questions"


            IF q IS last question THEN
                DISPLAY "Challenge question!"

                CALL create_question(var_max_num, var_max_num * 2) AS question

            ELSE
                CALL create_question(var_max_num / 2, var_max_num) AS question

            END IF


            CALL ask_question(question) AS (correct, seconds)
```

```
    IF correct THEN

        SET points TO MAX(1, 10 - seconds)

        INCREMENT var_score BY points

        INCREMENT var_correct_count BY 1

        DISPLAY "Correct! You answered in seconds second(s) - points point(s) awarded."

    ELSE

        DISPLAY "Incorrect! You answered in seconds second(s) - no points awarded."

    END IF


        APPEND (correct, seconds) TO var_results

    END FOR


    CALL show_results()
END METHOD


METHOD show_results

    CALCULATE percentage AS (var_correct_count / var_questions) * 100

    CALCULATE avg_time AS average of all times in var_results


    DISPLAY "Results:"

    DISPLAY final score, correct answers %, average response time


    IF var_correct_count EQUALS var_questions THEN

        DISPLAY "You're a Maths Master!"

    END IF
```

```
        DISPLAY "Breakdown:"

        DISPLAY table headers "Question  Correct  Time"

        FOR EACH (c, t) IN var_results WITH INDEX i

            DISPLAY i, "Yes" IF c ELSE "No", t seconds

        END FOR

    END METHOD


END CLASS


MAIN PROGRAM

    CREATE instance game OF MathsTest

    CALL game.play()

END MAIN


END MathsTest Program
```