Define "random\_list" function (receives "quantity", "minimum" and "maximum" parameters)

Set "result" to an empty list

Repeat “quantity” times

Generate a random “number” between "minimum" and "maximum"

Add “number” to "result" list

Return "result"

##############################################

Define “get\_question” function (receives “number” and “randomList” parameter)

Set “question” to an empty string

If number = 1

Set “question” to “What is the smallest number in this list?”, followed by “randomList”

Else if number = 2

Set “question” to "What is the biggest number in this list?”, followed by “randomList”

Else if number = 3

Set “question” to "What is the sum of the numbers in this list?”, followed by “randomList”

Else

Set “question” to "What is the average of the numbers in this list?”, followed by “randomList” and “(round UP to nearest integer)”

Return “question”

##########################################

Define “get\_correct\_answer” function (receives “number” and “randomList” parameters)

Set “answer” to None

If number = 1

Set “answer” to minimum value in “randomList”

Else if number = 2

Set “answer” to maximum value in “randomList”

Else if number = 3

Set “answer” to sum of the numbers in “randomList”

Else

Set “answer” to average of the numbers in “randomList” and round up to nearest integer

Return “answer”

########################################

Define “is\_valid” function (receives “value” parameter)

If uppercase “value” is in [“E”, “M”, “H”] list

Return True

Return False

#####################################

Define “get\_difficulty\_ attributes” (receives “difficulty” parameter)

If difficulty = “E”

Return [2, 3, 1, 5]

Else if difficulty = “M”

Return [4, 5, 3, 12]

Return [6, 8, 10, 25]

####################################

Define “print\_selcted\_difficulty” (receives “difficulty” parameter)

Set “text” to an empty string

If difficulty = “E”

Set “text” to “Easy”

Else if difficulty = “M”

Set “text” to “Medium”

Else

Set “text” to “Hard”

Print “text” concatenated with “difficulty selected!”

#####################################

Print welcome message

Prompt user to select a “difficulty”

Endless Loop

If the result of function *“is\_valid(difficulty)”* is True

Use function *“print\_selcted\_difficulty(difficulty)”* to print confirming message

Break out of loop

Print invalid message and re-prompt for “difficulty”

Set “questions”, “quantity”, “minimum” and “maximum” to the result of function *“get\_difficulty\_attributes(difficulty)”*

Set “score” to 0

For each question

Print question the user is up to out of the total number of “questions”

If the user is not up to the final question

Use function *“random\_list()”* to generate a list of "quantity" random numbers between "minimum" and "maximum"

Assign the result to “randomList”

Else

Print "Challenge questions!" message

Use function *“random\_list()”* to generate a list of "quantity" random numbers between double the "minimum" and double the "maximum"

Assign the result to “randomList”

Generate a random “number” between 1 and 4

Use function *“get\_question(number, randomList)”* to choose which question is asked

Print the question

Use function *“get\_correct\_answer(number, randomList)”* to obtain the correct answer for selected question

Assign the result to “correctAnswer”

Prompt for “answer”

If “answer” = “correctAnswer”

Print “Correct!” message

Else

Print “Incorrect” message, followed by “correctAnswer”

Print "Test complete!" message, followed by the "score" out of “questions” and the percentage that represent the “score”

If “score” = “questions”

Print "Perfect score, well done!" message