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" (inclusive), store as “number”

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 “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, set input to uppercase and store as “difficulty”

Endless Loop

If *“is\_valid(difficulty)”* returns True

Call the function *“print\_selcted\_difficulty(difficulty)”*

Break out of loop

Prompt for “difficulty” again

Call the function *“get\_difficulty\_attributes(difficulty)”* and store the returned value in “attributes”

Assign first element of “attributes” to “questions”

Assign second element of “attributes” to “quantity”

Assign third element of “attributes” to “minimum”

Assign fourth element of “attributes” to “maximum”

Set “score” to 0

For each question

Set “randomList” to None

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

If current question is not final question

Call the function *“random\_list(quantity, minimum, maximum)”* and assign the returned value to “randomList”

Else

Print "Challenge questions!"

Call the function *“random\_list(quantity, minimum \* 2, maximum \* 2)”* and assign the returned value to “randomList”

Generate a random number between 1 and 4, store as “number”

Call the function *“get\_question(number, randomList)”* and print the returned value

Call the function *“get\_correct\_answer(number, randomList)”*  and assign the returned value to “correctAnswer”

Prompt user to enter answer, store as “answer”

If “answer” = “correctAnswer”

Print “Correct!”

Add 1 to “score”

Else

Print “Incorrect”, followed by “correctAnswer”

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

If “score” = “questions”

Print "Perfect score, well done!" message