**1. Make a code that takes any list of numbers and calculates and displays the mean, median, standard deviation, variance, range and interquartile range. (Pseudocode, without using predefined functions)**

// Display the instructions to the user

Display "Enter a list of numbers, then press enter"

// Allow the user to enter a list of numbers, which is stored in the variable ‘list’

list = input

//initialize count to zero, which will be used to keep track of the number of elements in the list

Initialize count to zero

// Count the number of elements in the list

FOR each element in list

Read count

Add 1 to count

ENDFOR

Sort the elements in the list in ascending order

Set the sorted list to the variable sortedList

// CALCULATING THE MEAN

Initialize sum to zero

FOR each number in list: // calculates the sum of the elements in the list

Read sum

Add number to sum

ENDFOR

// The mean is found by dividing the sum of the elements by the total by the number of elements in the list

Set mean to sum / count

Display “The mean is: “ mean

//FINDING THE MEDIAN

IF count is odd

Set median to the middle element

ELSE // if count is even

Set median to the average of the two middle elements

ENDIF

Display “The median is:” median

// CALCULATING THE VARIANCE

// Initialize the sum of the squared deviations to zero

Initialize deviationsSumSquared to zero

// calculate the sum of the squared deviations from the mean

FOR each number in list

Set deviation to (number – mean)

Set squaredDeviation to (deviation \* deviation)

deviationsSumSquared = deviationsSumSquared + squaredDeviation

ENDFOR

// Calculate the variance by dividing the sum of the squared deviations by count

Set variance to deviationsSumSquared / count

Display “The variance is: ” variance

//CALCULATING THE STANDARD DEVIATION: This is the square root of the variance

//Set std to the square root of variance

std = variance ^ 0.5

Display "The standard deviation is: " std

//FINDING THE RANGE

//use the sorted list to identify the maximum and minimum numbers in the list

Set maxNumber to last number in sortedList

Set minNumber to first number in sortedList

Set range to maxNumber - minNumber

Display “The range is:” range

//CALCULATE INTERQUARTILE RANGE

Set Q1Index to (count + 1) \* 0.25 //Calculates the index of the 25th percentile

Set Q1 to the element at Q1Index in the sortedList //first quartile

Set Q3Index to (count + 1) \* 0.75 //Calculates the index of the 75th percentile

Set Q3 to the element at Q3Index in sortedList //third quartile

IQ3 = Q3-Q1

Display “The interquartile range is:” IQ3

**2. Create a code using pseudocode that randomly generates passwords of alternating letters and numbers of a user defined length**

//Prompt the user to enter his/her desired length of the password

Display "Enter the desired length of the password, (should be greater than 4) and press enter"

//Get input from the user and store the desired length in the variable length

Input length

//Define a function that takes ‘length’ as an argument, returns an empty string if condition is not me

FUNCTION passwordGenerator(length)

//Check if the length is less than or equal to 4

IF length > 0

Display "The length of the password should be greater than 0"

return ""

ENDIF

//initialize ‘password’ as an empty string variable that will store the generated password

Set password = ""

//initialize the letters and numbers that will be used in generating the password

Set numbers = "0123456789"

Set letters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ”

//use a while loop to generate the password until the desired length is reached

WHILE length > 0

//select a random letter from letters

Set letterIndex to a random number between 0 and 51

//add the letter at the letterIndex (the number selected) to password

Append letters[letterIndex] to password

//decrease the length of password by one as one character has been added

Subtract 1 from length

// if there are more characters to add, append a number to the password

IF length > 0

Set numberIndex to a random number between 0 and 9

//number at selected index added to password

Append numbers[numberIndex] to password

Subtract 1 from length

ENDIF

ENDWHILE

Return password

END FUNCTION

Display passwordGenerator(length) //Call the function and display the generated password

**4. Write a function using pseudocode that computes the list of the first 100 fibonacci numbers.**

//Create a function that returns a list of the first 100 Fibonacci numbers

FUNCTION cumputeFibonacci()

//initialize an empty list that stores the first Fibonacci numbers

fibList = []

//add 0 and 1, the first and second numbers, to the list

Set firstNumber to 0

Set secondNumber to 1

Append firstNumber to fibList

Append secondNumber to fibList

//iterates from 3 to 100, generating the next 98 Fibonacci numbers

FOR number = 3 to 100

// Calculate the next Fibonacci number

Set nextNumber = firstNumber + secondNumber

Append nextNumber to fibList

//update firstNumber and secondNumber to the most recent numbers

firstNumber = secondNumber

secondNumber = nextNumber

ENDFOR

// Return the list of Fibonacci numbers

Return fibList

END FUNCTION

// Call the function and display the first 100 Fibonacci numbers

Display computeFibonacci()

**5. Write a single function using pseudocode that takes an input temperature in Celsius, converts it from Celsius to Fahrenheit, Celsius to kelvin and Fahrenheit to kelvin and shows the result**

//take the temperature in Celsius from the user and assign it to tempCelsius

Display “Enter the temperature in Celsius and press enter”

tempCelsius = input()

Display "Temperature in Celsius: " tempCelcius

// Function convertTemp takes tempCelsius as input and performs the various conversions

FUNCTION convertTemp(tempCelsius)

//converting from Celsius to Fahrenheit

tempFahrenheit = (tempCelcius \* 9/5) + 32

Display "Temperature in Fahrenheit: " tempFahrenheit

//converting from Celsius to Kelvin

tempKelvin = tempCelcius + 273.15

Display "Temperature in Kelvin: " tempKelvin

//Converting from Fahrenheit to kelvin

kelvinFromFahrenheit = ((tempFahrenheit - 32) \* 5/9) + 273.15

Display "Temperature in kelvin from Fahrenheit: " + kelvinFromFahrenheit

END FUNCTION

// Call the function

convertTemp(tempCelsius)

6. Start a Linux session and do the following.

1. Determine where you are

A black and purple rectangle

Description automatically generated with medium confidence

1. Make a directory

A black and purple rectangle

Description automatically generated

1. Go to the directory

A black and purple rectangle

Description automatically generated

1. Make a file called hw1.o

A screenshot of a computer program

Description automatically generated

1. Copy hw1.o to test.txt

A screenshot of a computer program

Description automatically generated

1. What files are in this directory, and which was generated first

A screenshot of a computer

Description automatically generated

file hw1.o was generated first.

1. Display the top of both files

A screenshot of a computer program

Description automatically generated

1. Return to the home directory

A computer screen shot of a computer

Description automatically generated