Software Development Challenge

Overview

The pseudo code below is submitted as part of the **General Assembly Software Development Challenge**. This document is also available on **GitHub**.

Requirement

Using a language of your choice write a program which takes in the input of a comma delimited text string containing a series of three or more increasing integers. Your program should determine a pattern being exhibited and output the next 10 integers in the series.

Run your program using the sets [1, 9, 25, 49] and [7, 12, 17] and submit an archive containing both the documented source code and the output.

Pseudo Code

```
//solve for equations of type Y = Ax^2+Bx+C
// 0 = ZERO in statements
// lowercase denotes variable or property
// LESS THAN denotes operator
// = denotes assignment, EQUALS logical test
// [] denotes array, () denotes function

receive input series
reset outputs
SET A = 0
SET B = 0
SET C = 0

SET series[] = seriesInput

IF validateInput() EQUALS TRUE
        IF findSeries() EQUALS TRUE
                SWITCH seriesType
                        seriesType = secondOrder
                                FORMAT secondOrder output
                                PRINT secondOrder output
```

```
            seriesType = firstOrder
                FORMAT firstOrder output
                PRINT firstOrder output
            ELSE
                FORMAT errorMessage
                PRINT errorMessage


FUNCTION findSeries()

    IF seriesIsFirstOrder() EQUALS TRUE RETURN TRUE
    IF seriesIsSecondOrder() EQUALS TRUE RETURN TRUE

FUNCTION seriesIsFirstOrder

    SET result = FALSE
    SET firstOrderGradient[] = getDeltasBetweenPoints(series[])
    IF firstOrderGradient [1] MINUS firstOrderGradient [0] EQUALS 0

        SET result = TRUE

    IF result EQUALS TRUE

        SET seriesType = firstOrder
        SET B = firstOrderGradient [0]
        SET C = series[0] MINUS B
        RETURN result

    ELSE

        RETURN result

FUNCTION seriesIsSecondOrder

    SET result = FALSE
    SET firstOrderGradient[] = getDeltasBetweenPoints(series[])
    SET secondOrderGradient[] =
```

```
                getDeltasBetweenPoints(firstOrderGradient[])

SET A = secondOrderGradient[1] divided by 2
SET C = series[0]

SET diffs[] = new empty array

FOR B = 0 TO B less than 3 INCREMENT B
        SET testValue[] = new empty array
        FOR X = 0 TO X LESS THAN series[] length INCREMENT X
                SET testValue[X] = A*(X*X)+(B*X)+C

        SET diffs[] = getDifferences(B,testValue[],series[])

SET bIsPositive = FALSE

IF diffs[2][2][1] LESS THAN diffs[2[1][1]
        AND diffs[2][2][1] less than diffs[1][1][1]

                SET bIsPositive = TRUE

SET B = 0
SET bStart = 0
SET DIFFS = new Array

WHILE result = FALSE

        FOR X = 0 TO X LESS THAN series[] length MINUS 2
        INCREMENT X
                SET testValue[X] = A*(X*X)+(B*X)+C

        SET diffs[] = getDifferences(ABS(B),testValue[],series[])

        FOR X = 0 TO X LESS THAN series[] length INCREMENET X
                IF diffs[ABS(B)][X][1] EQUALS 0 AND diffs[ABS(B)][X+1]
                EQUALS 0
                        SET RESULT = TRUE
```

```
            IF RESULT EQUALS FALSE
                IF bIsPositive EQUALS TRUE
                    INCREMENT B By 1

                ELSE
                    DECREMENT B by 1

        IF RESULT EQUALS TRUE

            SET seriesType = 2
            SET B = B
            RETURN RESULT

        ELSE

            RETURN RESULT


FUNCTION validateInput

        IF seriesInput length GREATER THAN 0

            REPLACE spaces in seriesInput with empty string

            IF seriesInput CONTAINS comma

                SET series[] = SPLIT seriesInput ON comma

                IF series[] length EQUALS 0 or TYPEOF series EQUALS
                UNDEFINED

                    RETURN FALSE

                ELSE

                    IF series[] length IS LESS THAN 3

                        RETURN FALSE
```

```
                    ELSE

                        RETURN TRUE
            ELSE

                    RETURN FALSE
        ELSE

                    RETURN FALSE
```

```
FUNCTION getDifferences(iteration,sample[],actual[])

    SET deltas[] = new array

    FOR X = 0 to X LESS THAN series[] length INCREMENT X

        SET deltas[X] = new
            Array (iteration,sample[x] MINUS actual[x])

        RETURN deltas[]

FUNCTION getGradient(series)

    SET deltas[] = new array

    FOR X = 0 to X LESS THAN series[]  length MINUS 1
    INCREMENT X

        SET deltas[X] = series[X+1] MINUS series[X]

    RETURN deltas[]
```