11 Students in a class are recording the amount of time in minutes spent in front of a screen for each day of the week. x5 (x7) int

The one-dimensional (1D) array `StudentName[]` contains the names of the students in the class.

Bob
Bert
Jim
⋮

Class Size

days of week

The two-dimensional (2D) array `ScreenTime[]` is used to input the number of minutes on each day spent in front of a screen. [row, col]

Bob
10, 4, 100

The position of each student's data in the two arrays is the same. For example, the student stored at index 10 in `StudentName[]` and `ScreenTime[]` is the same.

The variable `ClassSize` contains the number of students in the class.

DECLARE StudentName : ARRAY[1:ClassSize] OF STRING

Write a program that meets these requirements:

- ✓ allows all the students to enter their daily minutes of screen times for the past week
- ✓ calculates the total number of minutes of screen time for each student in the week
- ✓ counts, for each student, the number of days with more than 300 minutes of screen time
- ✓ calculates the average weekly minutes of screen time for the whole class
- ✓ finds the student with the lowest weekly minutes of screen time
- outputs for each student:
  - ✓ name .
  - ✓ total week's screen time in hours and minutes
  - ✓ number of days with more than 300 minutes of screen time
- ✓ outputs the average weekly minutes of screen time for the whole class
- ✓ outputs the name of the student with the lowest weekly screen time.

inside FOR s_counter loop

You must use pseudocode or program code **and** add comments to explain how your code works. All inputs and outputs must contain suitable messages.

Assume that the array `StudentName[]` and the variable `ClassSize` already contain the required data.

You do **not** need to declare any arrays or variables; you may assume that this has already been done.

```
DECLARE Total : INTEGER
ClassTotal ← 0
Lowest_Minutes ← 99999
FOR s_counter ← 1 TO ClassSize    // loop for each student
    Total ← 0        Day_over_300 ← 0
    FOR day_counter ← 1 TO 5
    REPEAT
        OUTPUT "Please enter minutes for day", daycounter
        INPUT Minutes
    UNTIL Minutes >=0
        ScreenTime [s_counter, day_count] ← Minutes
        Total ← Total + Minutes    // running total for student
        IF Minutes > 300 THEN
            Day_over_300 ← Day_over_300 +1
        END IF
    NEXT day_counter
IF Total < Lowest_Minutes THEN
    Lowest_Minutes ← Total
    LowestStudent ← s_counter
```