Question	Answer	Marks
11	Check if each requirement listed below has been met. Requirements may be met using a suitable built-in function from the programming language used (Python, VB.NET or Java). On the script, add seen if the requirement has been met, NE if a partial attempt, or a cross if no attempt. Use the tables for AO2 and AO3 below to award a mark in a suitable band using a best fit approach, then add up the total. Marks are available for: • AO2 (maximum 9 marks) • AO3 (maximum 6 marks) **Data structures required** with names as given in the scenario: Arrays or lists `StudentName`, `ScreenTime` Variable `ClassSize` could be constant **Requirements (techniques)** **R1** allows a student to enter their weekly screen time and calculates the total number of minutes of screen time for each student in the week (input, iteration and totalling) **R2** counts the number of days with more than 300 minutes screen time each day and calculates the average week's screen time for the whole class (selection, counting, iteration, calculating average) **R3** finds the student with the lowest weekly minutes. Outputs for each student: name, total week's screen time in hours and minutes, number of days with more than 300 minutes screen time, outputs the average weeks screen time for the whole class and the name of the student with the lowest number of minutes (finding minimum value, output)	**15**

Question	Answer	Marks
11	***Example 15-mark answer in pseudocode***	

```
WeekLength ← 5
LowestMinutes ← 1000
ClassTotal← 0
FOR StudentCounter ← 1 to ClassSize // loop for each student
    Total ← 0
    DaysOver300 ← 0
    FOR DayCounter ← 1 to WeekLength // loop for each day
        REPEAT
            OUTPUT "Please enter number of minutes for day ", DayCounter
            INPUT Minutes
        UNTIL Minutes >= 0
        ScreenTime[StudentCounter,  DayCounter] ← Minutes
        Total ← Total + Minutes
        IF Minutes > 300
          THEN
            DaysOver300 ← DaysOver300 + 1
        ENDIF
        IF Minutes < LowestMinutes
          THEN
            LowestMinutes ← Minutes
            LowestIndex ← StudentCounter
        ENDIF
    NEXT DayCounter
    OUTPUT StudentName[StudentCounter]
    OUTPUT "Screen time ", DIV(Total, 60), " hours ", MOD(Total, 60), " minutes "
    OUTPUT "Days with more than 300 minutes screen time ", DaysOver300
    ClassTotal ← ClassTotal + Total
NEXT StudentCounter

OUTPUT "Average weekly screen time for class ", ClassTotal / ClassSize, " minutes "
OUTPUT "Lowest weekly time ", StudentNames[LowestIndex]
```

Marking Instructions in italics			
AO2: Apply knowledge and understanding of the principles and concepts of computer science to a given context, including the analysis and design of computational or programming problems			
0	**1–3**	**4–6**	**7–9**
No creditable response.	At least one programming technique has been used. *Any use of selection, iteration, counting, totalling, input and output.*	Some programming techniques used are appropriate to the problem. *More than one technique seen applied to the scenario, check the list of techniques needed.*	The range of programming techniques used is appropriate to the problem. *All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check the list of techniques needed.*
	Some data has been stored but not appropriately. *Any **use** of variables or arrays or other language dependent data structures e.g. Python lists.*	Some of the data structures chosen are appropriate and store some of the data required. *More than one data structure **used** to store data required by the scenario.*	The data structures chosen are appropriate and store all the data required. *The data structures **used** store all the data required by the scenario.*

Marking Instructions in italics			
AO3: Provide solutions to problems by: **evaluating computer systems**		**making reasoned judgements**	**presenting conclusions**
0	**1–2**	**3–4**	**5–6**
No creditable response.	Program seen without relevant comments.	Program seen with some relevant comment(s).	The program has been fully commented
	Some identifier names used are appropriate. *Some of the data structures used have meaningful names.*	The majority of identifiers used are appropriately named. *Most of the data structures used have meaningful names.*	Suitable identifiers with names meaningful to their purpose have been used throughout. *All of the data structures used have meaningful names.*
	The solution is illogical.	The solution contains parts that may be illogical.	The program is in a logical order.
	The solution is inaccurate in many places. *Solution contains few lines of code with errors that attempt to perform a task given in the scenario.*	The solution contains parts that are inaccurate. *Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.*	The solution is accurate. *Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.*
	The solution attempts at least one of the requirements. *Solution contains lines of code that attempt at least one task given in the scenario.*	The solution attempts to meet most of the requirements. *Solution contains lines of code that attempt most tasks given in the scenario.*	The solution meets all the requirements given in the question. *Solution performs all the tasks given in the scenario.*