**Digital Technologies Department**

**91076 Coding Assessment (Internal)**

Context and issue

A researcher is collecting and analysing data on TV viewing time for various people. Design a program to determine and print out someone’s name and their average weekly TV viewing time. Data entered is the name followed by the 7 recorded viewing times for the week. Also determine and printout the highest recorded viewing time for that person. For example, if the data entered is:

Wiremu Jones, 3, 1.2, 0, 4, 3.7, 6, 1.4

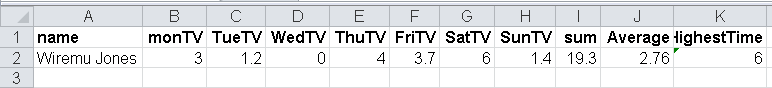
Output is:

Wiremu Jones

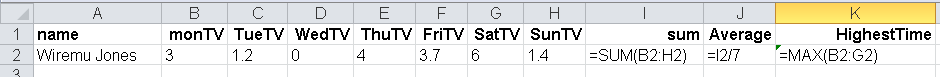
Average viewing time: 2.76 hours

Highest viewing time: 6 hours

Test Data of Expected inputs

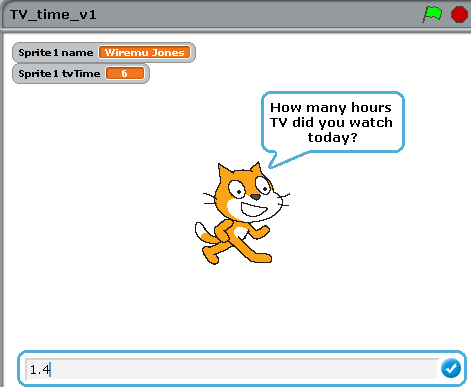


Formulas

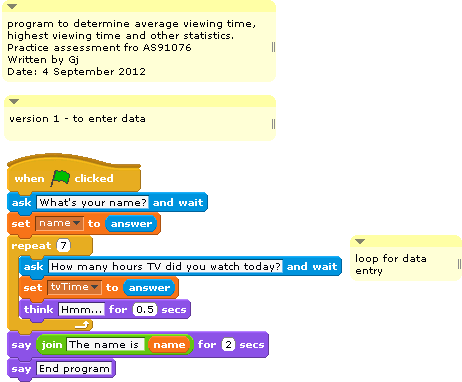


**Version 1 of Plan**

Output



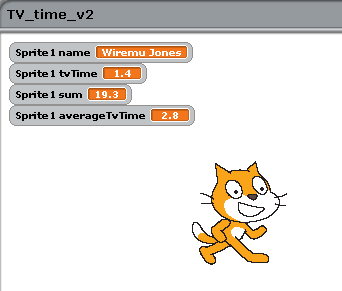
Data is entered in loop

Code for Version1

Put in statement for processing - changed time to 0.5 from 2 sec as took too long

**Version 2 of Plan**

Output

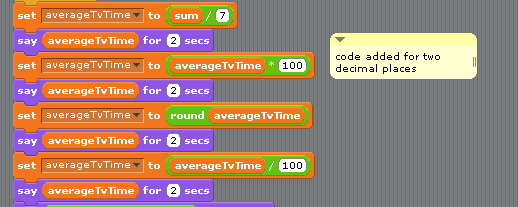


Correct values for sum and average - but not two decimal places for average

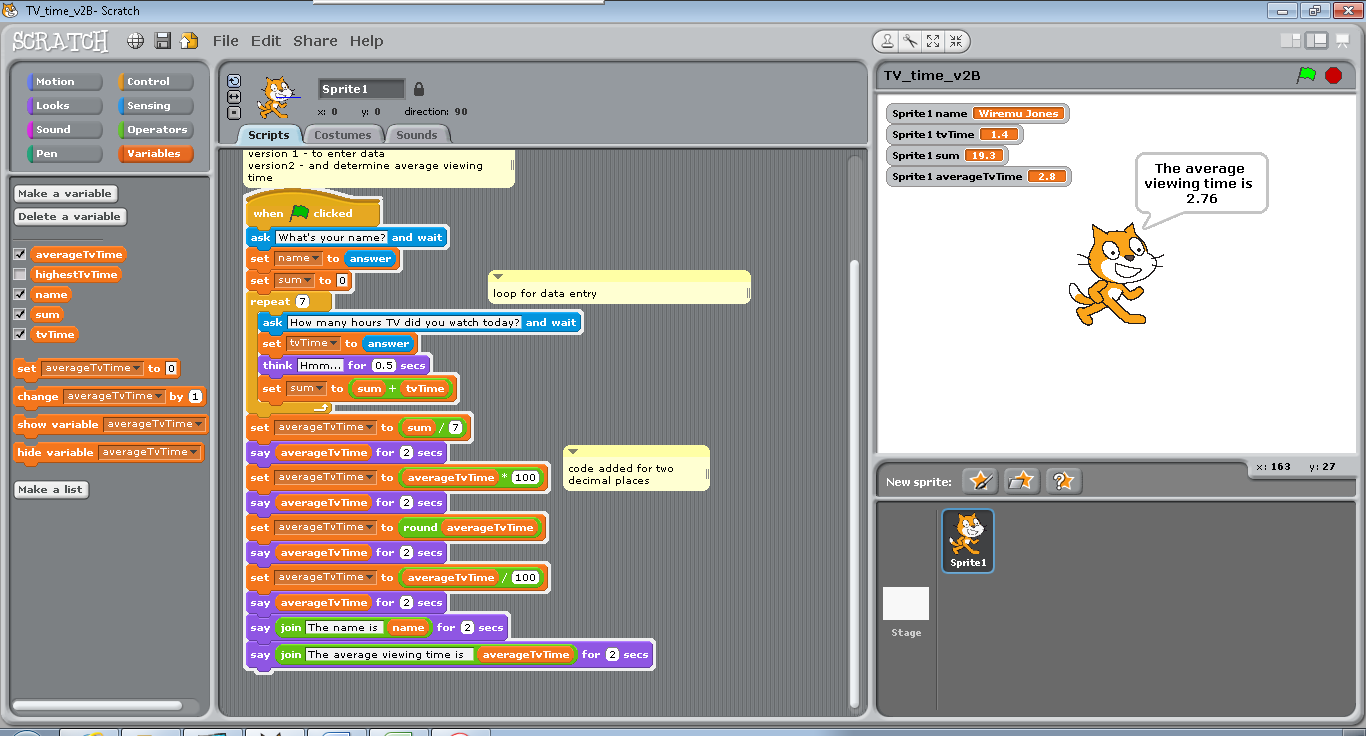
Error must be in this statement



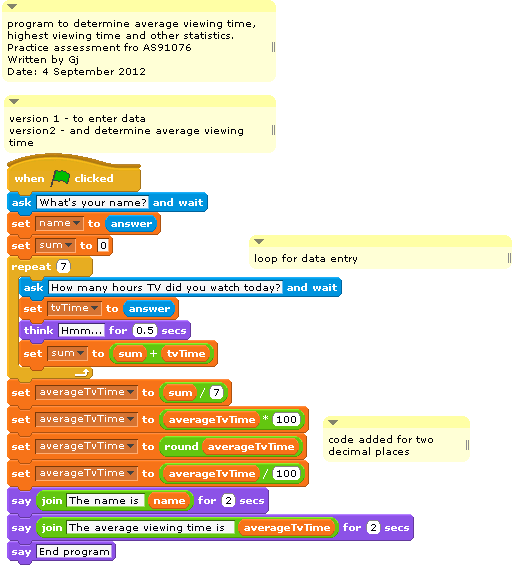
Added in following lines of code with print statements to check calculations



Now getting the correct answers, even though memory storage showing one decimal place

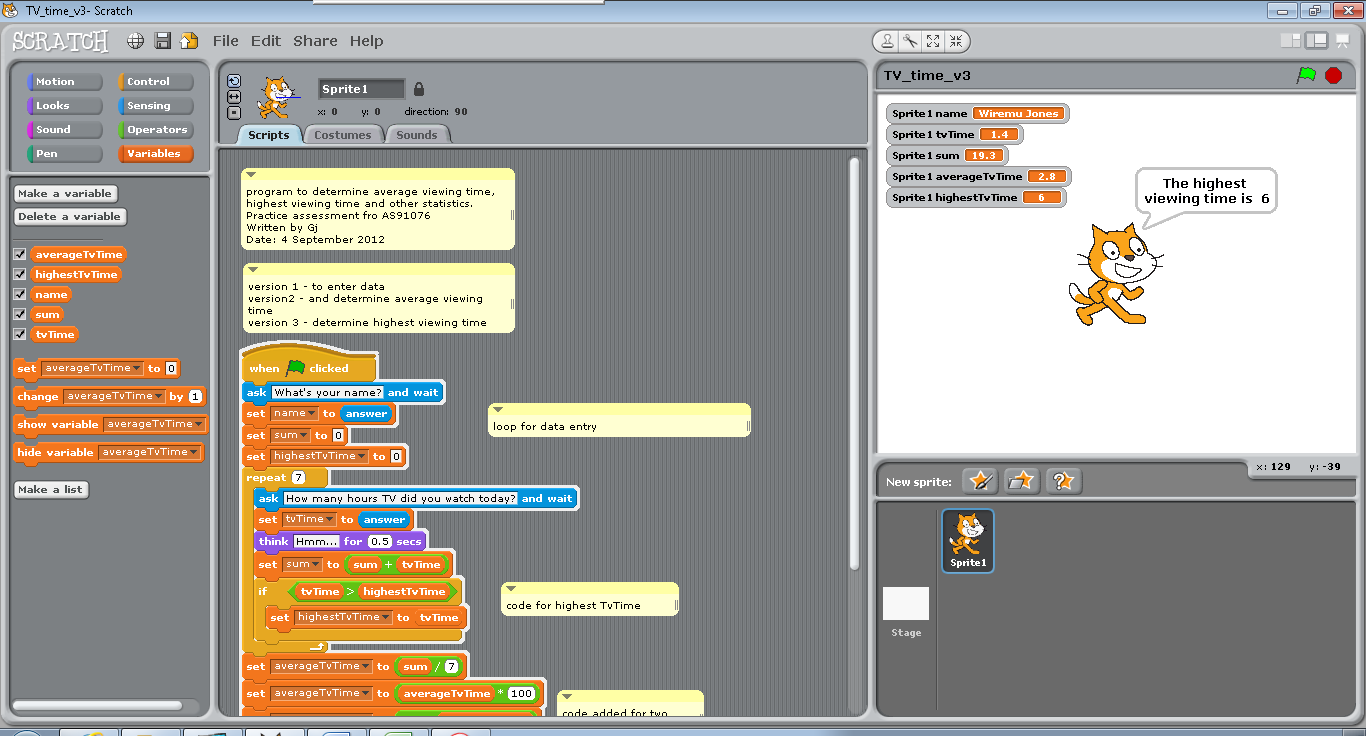


Tidied code for version 2 - without extra print/display statements



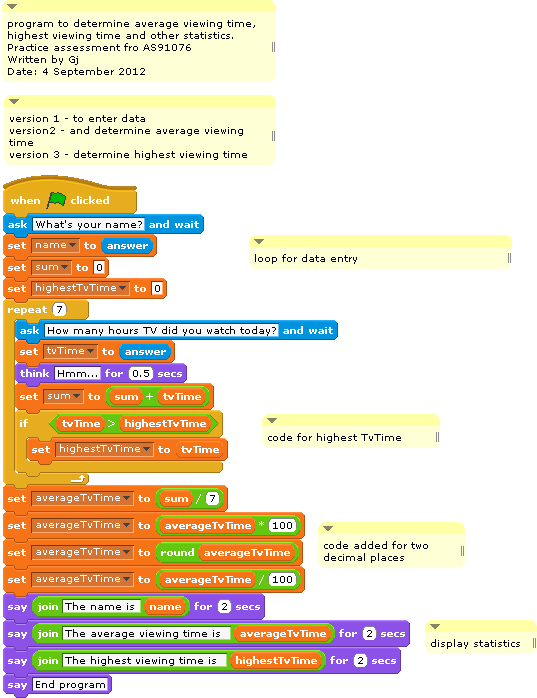
**Version 3 of plan**

Output



Output is correct with expected test data

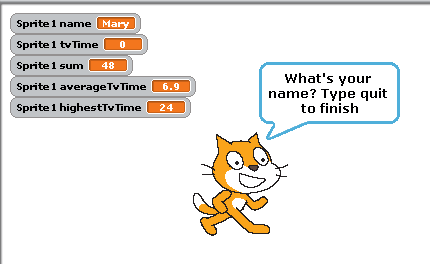
Code for version 3



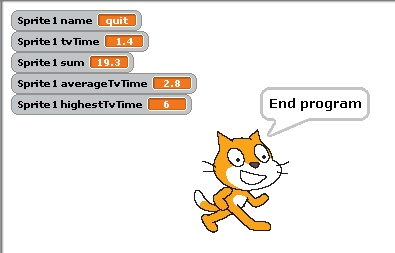
**Testing Table for Version 4**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Data | What is being tested | Expected output | Actual Output |
| Mary, 0, 24,0,0,0,0,24 | Boundary of limits for hours | Average viewing time 6.86  Highest viewing time 24 |  |
| Name of quit | Exceptional No data entered | No data entered |  |

Output for boundary limits



Output correct as from above table. Boundary data accepted



Name of quit stops program with no new results

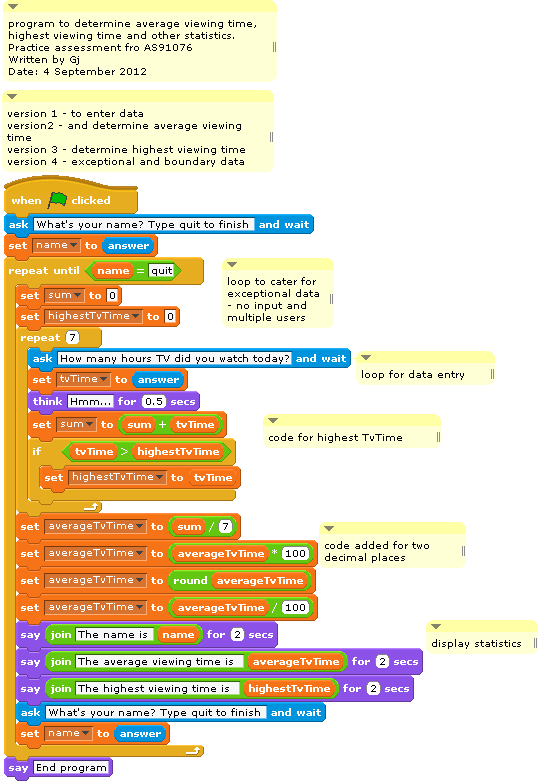
Memory locations keep previous run answers

First version of code for merit

Is looking at multiple users and program is flexible for more than one user, as well as being robust if no data entered.

Boundary and expected data still give output with no unexpected behaviours.

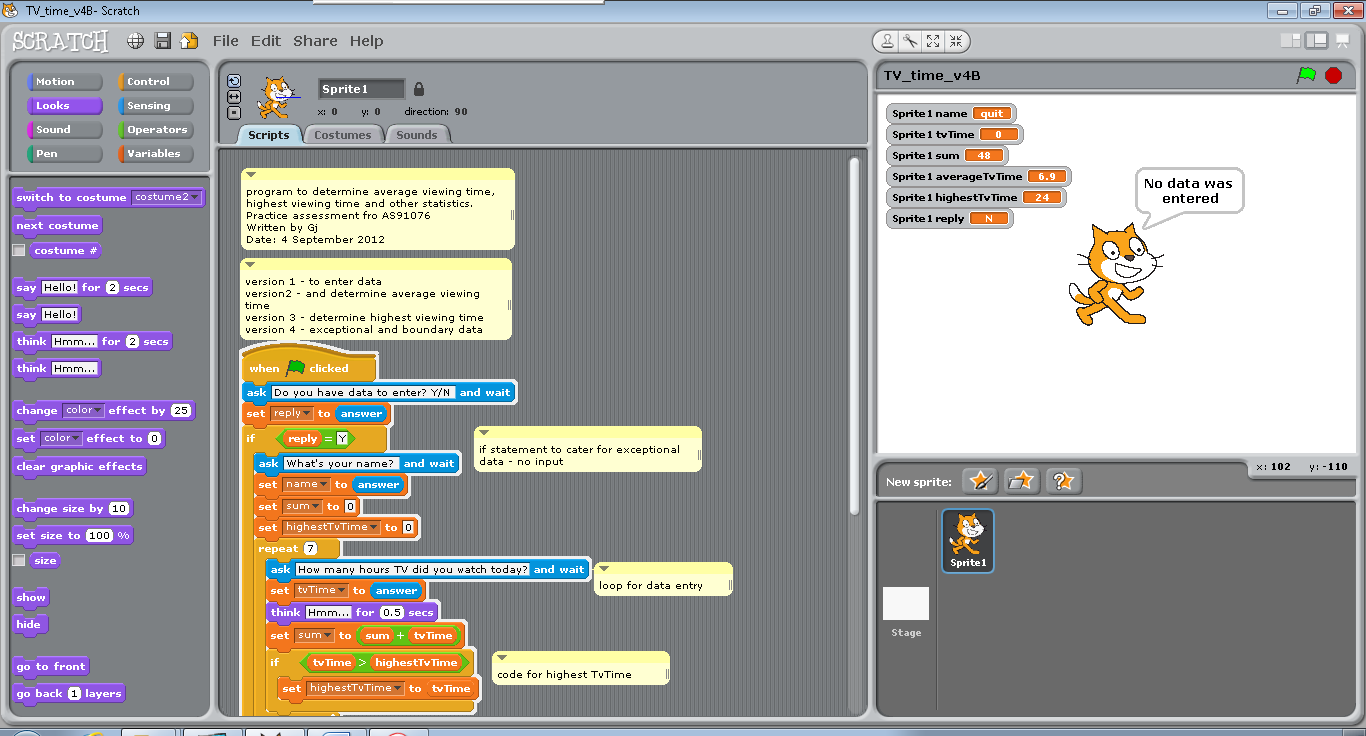
Could also use a flag to give a suitable message of no data entered.



Second version of code for merit.

Will be robust enough to cope with no data entered and boundary values

Output shows suitable message for no data entered

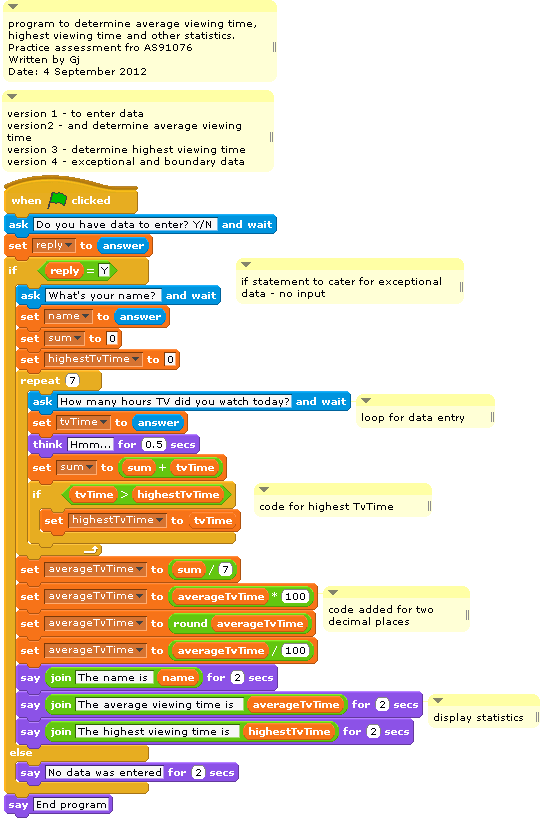


Alternative code for merit solution - Version 4B

(Note for marker and students:

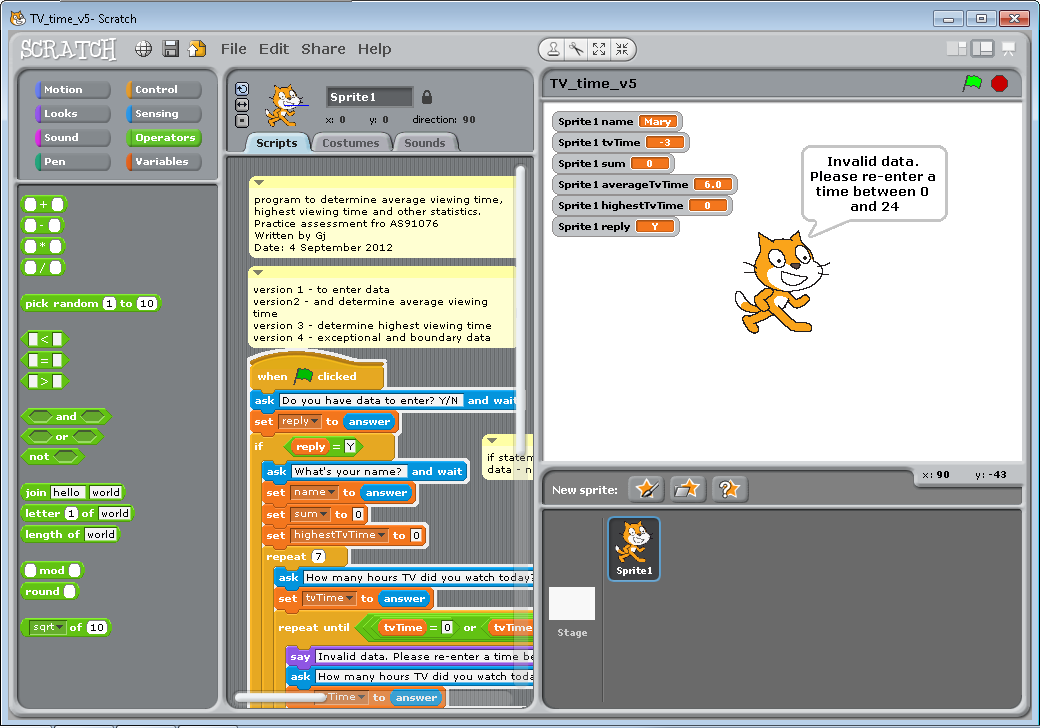
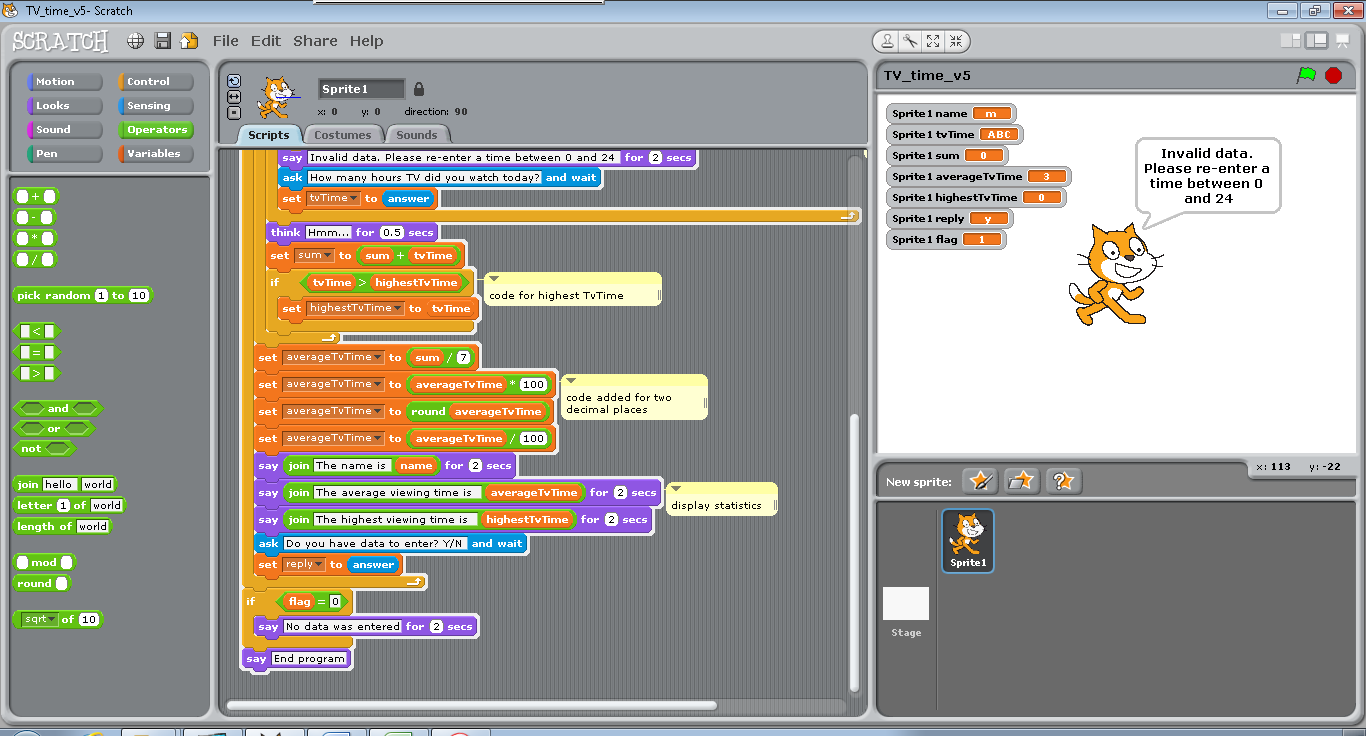
An if statement is sufficient in this case for no data entered. The version above for multiple users , is bordering on excellence for a robust and flexible program.

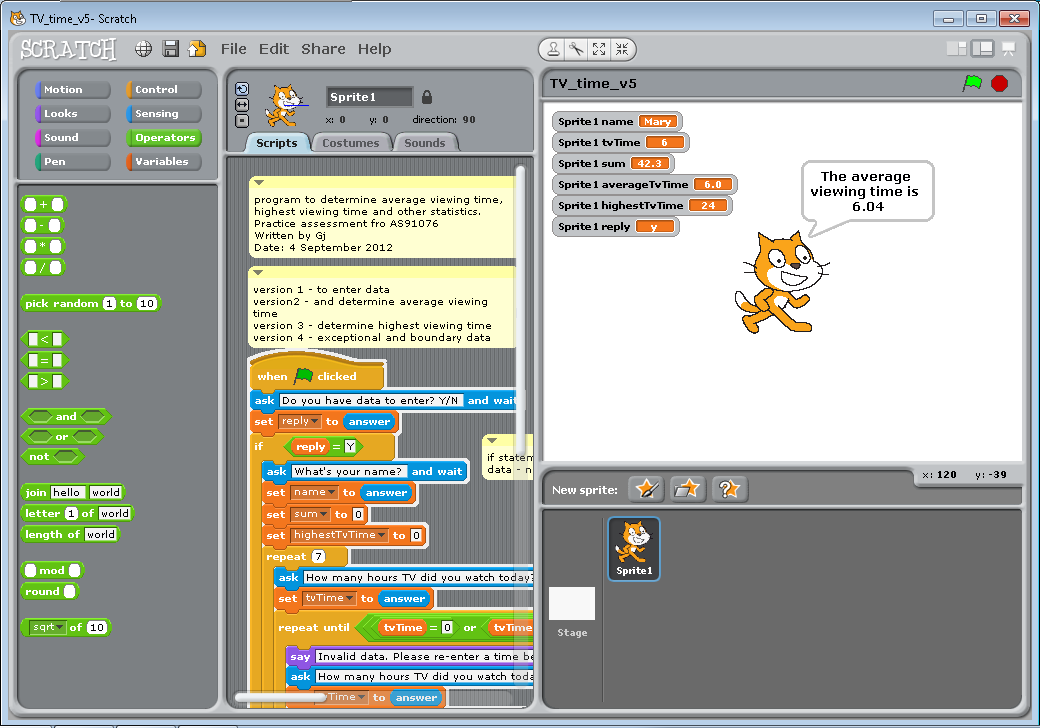
But without checks for data validity - is still a merit solution)



**Testing table for invalid data - version 5**

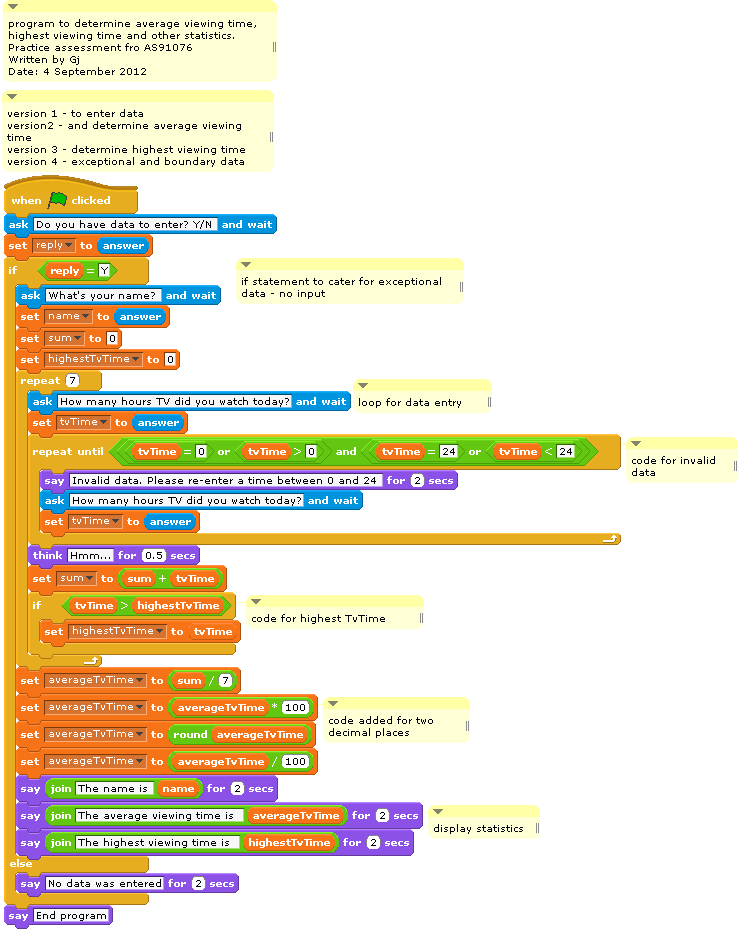
|  |  |  |  |
| --- | --- | --- | --- |
| Test Data | What is being tested | Expected output | Actual Output |
| Mary, -3, 0, 26, 24, 2.3, 1, 4, 5, 6 | Invalid data for out of range values for time | Error message and re-entry of correct data  Average viewing time 6.04  Highest viewing time 24 |  |
|  | | | |
| ABC in place of numbers for time | Invalid data input | Error message and correct data |  |





Correct data and output and error message for invalid data.

Code for version 5



Alternative form of version 5 - with multiple users for a more flexible and robust program

(Note to assessor and students. Above version would also get excellence)

This version has been comprehensively tested with expected, exceptional, boundary and invalid data.

