# Marking

## Algorithms and Code (1 mark for each)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pseudocode | Flowchart | Python |
| **main** |  |  |  |
| control the flow of the program |  |  |  |
| exit without crashing when appropriate |  |  |  |
| **loadArray** |  |  |  |
| prompt the user to enter a number |  |  |  |
| check that the entered value is valid |  |  |  |
| add the number to the array |  |  |  |
| return the program to the main routine |  |  |  |
| **printArray** |  |  |  |
| print out the values in the array |  |  |  |
| print out the total number of elements in the array |  |  |  |
| return the program to the main routine |  |  |  |
| **sumArray** |  |  |  |
| print out the sum total of the values in the array |  |  |  |
| print out the number of elements in the array |  |  |  |
| return the program to the main routine |  |  |  |
| **findMax** |  |  |  |
| print a message indicating has been called |  |  |  |
| return the program to the main routine |  |  |  |
| **findMin** |  |  |  |
| print a message indicating has been called |  |  |  |
| return the program to the main routine |  |  |  |

## Desk check (24 marks)

Each subroutine and main tested in a separate table

Each table tests:

* All boundary conditions
* All paths through the code
* Validity of entered data (ie includes data outside of the expected range)

## Comparing solutions (8 marks)

Compare your coded solution with the solution of another classmate. Comment in the:

* Similarities and differences
* Strengths and weakness of each approach.