

## Task 1: software design and development (part B)

The design for the Scottish Jumping Jacks finalists' program is shown below.

### Program top-level design (pseudocode)

- |   |   |
|---|---|
| 1. Get qualifying athletes' data  | OUT: entryID(), location(), forename(),<br>surname(), jumps() |
| 2. Generate bib values and write to new file with<br>entry IDs                                  | IN: entryID(), location(), forename(), surname()              |
| 3. Find the highest number of jumping jacks<br>completed  | IN: jumps(),<br>OUT: maxJumps                                 |
| 4. Display the full name of the athlete(s) who<br>completed the highest number of jumping jacks | IN: maxJumps, forename(), surname(), jumps()                  |

### Refinements

- 1.1 Open athletes.csv file
- 1.2 Loop for thirty athletes
- 1.3     Store entryID, location, forename, surname, jumps for athlete in parallel arrays
- 1.4 End loop
- 1.5 Close athletes.csv file
  
- 2.1 Create bibValues.csv file
- 2.2 Loop for thirty athletes
- 2.3     Set bibValue to first letter of forename & full surname & ASCII value of first letter of location
- 2.4     Write entryID and bibValue to file
- 2.5 End loop
- 2.6 Close bibValues.csv file
  
- 3.1 Set maximum jumps to the value stored in the first index of the jumps array
- 3.2 Start loop from second index to end of array
- 3.3     If the current number of jumps is more than maximum jumps then
- 3.4         Set maximum jumps to current number of jumps
- 3.5     End if
- 3.6 End loop
- 3.7 Return maximum jumps
  
- 4.1 Loop for thirty athletes
- 4.2     If current number of jumps equals maximum jumps then
- 4.3         Display forename and surname
- 4.4     End if
- 4.5 End loop

**1c(i)** Using the problem description and design, implement the program in a language of your choice. Your program should:

- ♦ be maintainable and modular
- ♦ use a function to find and return the maximum number of jumps
- ♦ follow the design and the refinements provided

**(13 marks)**

Print evidence of:

- ♦ your completed program code
- ♦ your output, showing athlete(s) with the maximum number of jumps
- ♦ your CSV file containing the entry ID and bib values

Include your name and candidate number on all evidence.

**1c(ii)** The location with the fewest number of athletes qualifying will host the next final.

A new sub-program is to be implemented to find the total number of athletes from each location in the final. An example of the output is shown below.

```
Coatbridge has 6 finalists
Inverness has 8 finalists
Kirkcaldy has 7 finalists
Motherwell has 9 finalists
```

Implement the additional sub-program.

**(2 marks)**

Print evidence of:

- ♦ your edited program code
- ♦ the display produced by the new sub-program

Include your name and candidate number on all evidence.

- 1d** The function to find the maximum number of jumps is tested using the following test data.

jumps = [100,87,102,108,95]

A watchpoint is placed on the variable storing the maximum number of jumps.

Complete the table below by entering:

- ♦ the lines of code from your program where the watchpoint is triggered
- ♦ the value of the maximum number of jumps variable when the watchpoint is triggered

**(3 marks)**

Line of code from your program	Value of the maximum number of jumps

- 1e** With reference to your own program code, evaluate:

**(2 marks)**

the fitness for purpose of the function to generate bib values
the maintainability of your program, referring to modularity

Candidate name\_\_\_\_\_ Candidate number\_\_\_\_\_