

Higher Computing Science

Assignment

Assessment task

This document provides information for teachers and lecturers about the coursework component of this course in terms of the skills, knowledge and understanding that are assessed. It must be read in conjunction with the course specification.

Valid for session 2020-21 only.

This assessment is given to centres in strictest confidence. You must keep it in a secure place until it is used.

Where practically possible, you should adhere to the published conditions of assessment for this assignment in order to maintain the reliability and validity of the assessment evidence generated.

This edition: January 2021 (version 1.0)

© Scottish Qualifications Authority 2021

Task 1: software design and development (part A)

Problem description

Thirty athletes have qualified for the final of the Scottish Jumping Jacks competition. Qualifying events were held at four locations, where each athlete performed as many jumping jacks as they could in 1 minute.

The following details are stored in a CSV file, for each athlete who qualified for the final:

- ◆ Entry ID
- ◆ Qualifying location
- ◆ Forename
- ◆ Surname
- ◆ Number of jumping jacks completed



Purpose

A program is required to read each of the athlete's data from the existing CSV file. A bib value for each of the finalists will be generated and stored in a new CSV file along with the entry ID. The program will also display the full name of the athlete(s) who completed the highest number of jumping jacks.

An example of the bib value generated for the athlete Daniel Currie, who qualified at the Inverness event, is shown below. Note that 73 is the ASCII value of 'I', the first character of Inverness.



Assumptions

- ◆ The CSV file has data for 30 athletes, is formatted correctly and is error-free.
- ◆ Each line of the CSV file stores the entry ID, qualifying location, forename, surname and number of jumping jacks completed at qualification, as shown below:

```
f01,Motherwell,Ellie,McAninch,85
f02,Inverness,Ayat,Whyte,83
f03,Kirkcaldy,Simra,Zamora,42
f04,Motherwell,Dai,Nguyen,37
f05,Coatbridge,Max,Hughes,113
...
```

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