Higher
Coursework
Assessment Task

Higher Computing Science
Assignment
Finalised Marking instructions

© Scottish Qualifications Authority 2022

Downloaded free from https://sqa.my/

Marking instructions

General marking principles

This information is provided to help you understand the general principles that must be applied when marking candidate responses in this assignment. These principles must be read in conjunction with the specific marking instructions, which identify the key features required in candidate responses.

a Marks for each candidate response must always be assigned in line with these general marking principles and the specific marking instructions for this assessment.

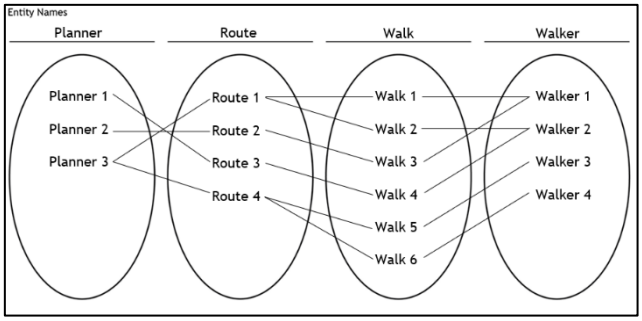
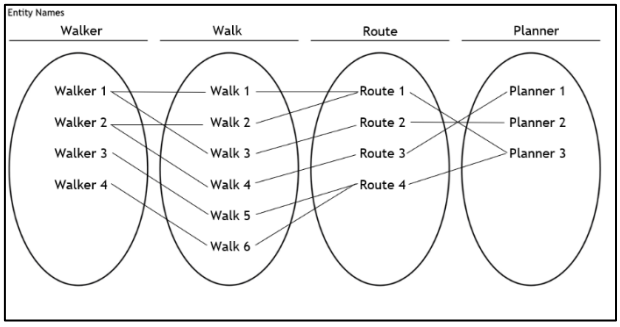
b Marking should always be positive. This means that, for each candidate response, marks are accumulated for the demonstration of relevant skills, knowledge and understanding: they are not deducted from a maximum on the basis of errors or omissions.

Specific marking instructions

Task	Expected response	Additional guidance	Marks available	
1	**Software design and development**			
1a	1 mark for each bullet. Max 2 marks. ♦ date is correct (UK) format (day/month) ♦ date is within an acceptable range e.g. no 31st September ♦ age is a whole number/integer ♦ age is within an acceptable range ♦ only one town and one mammal on each line	For two marks, must have two bullets from left. Award 1 mark for general reference to data that is valid/present/complete/formatted. Must specify town and mammal or state 4 values per line.	2	Analysis
1b	1 mark for each bullet. ♦ identify first character ♦ if not upper-case convert to upper-case ♦ concatenate with remaining string ♦ return string/variable/value	Accept reference to pre-defined functions in design for conversion. Return value should be referenced in previous steps.	4	Design
1c	Read in Mammals Data (2) ♦ module with correct parameter passed or returned to read data from file to array of records ♦ each line of sightings data stored in a record structure assigned to an array Find oldest person (3) ♦ module with correct parameter passed and max displayed within procedure ♦ initialise and re-assign max age ♦ if statement to find correct max	If candidate uses parallel arrays as a data structure, award 0 marks for "read" procedure then accept appropriate parameter passing for parallel arrays in the remainder of the procedures/functions. Award 0 marks for bullets 2 and 3 if a pre-defined function is used in implementation instead of a findinding maximum algorithm. Accept finding position of max value to produce output.	15	Implementation

Task	Expected response	Additional guidance	Marks available	
1	**Software design and development**			
	Upper-case function (3) ♦ extract first character ♦ if statement to convert to upper-case using ASCII/Char pre-defined function ♦ return original or concatenated string Display dates of sightings (2) ♦ module with correct parameter passed, correct use of the function and dates displayed within procedure ♦ linear search to find sighting dates Display the number of sightings each day (3) ♦ module with correct parameter passed, dayToCount (date) and count displayed within procedure ♦ count initialised to 1 and incremented for a single date ♦ dayToCount and count reset for each new date Implementation (2) ♦ a single upper-case function called twice ♦ modular (4 procedures and 1 function) and maintainable	Do not deduct marks for two separate functions for town and mammal at this step. Do not penalise if no step 4.12 or display for 30/09/2021. Maintainability should include evidence of meaningful identifiers, internal commentary, indentation and white space in the context of the program.		
1d	1 mark for each bullet. Max 2 marks. • watchpoint set on count variable • count increments by 1 (while date is 01/09/21) OR count is 6 when dayToCount changes to 02/09/21 OR • watchpoint set on dayToCount • count should be 6 when dayToCount moves to 02/09/21		2	Testing

Task	Expected response	Additional guidance	Marks available	
1	**Software design and development**			
1e	Efficiency 1 mark for any bullet. • A single function can be used to check first characters of town and mammal • First character only changes if found to be lower-case • Accept explanation for inefficient Maintainability 1 mark for any bullet. ♦ linking modularity to maintainability e.g. sub procedures can be edited independently ♦ local variables prevent clashes with variables in other parts of the code		2	Evaluation

Task	Expected response	Additional guidance	Marks available	
2	**Database design and development**			
2a	1 mark for each bullet. ♦ entity names, in the correct order ♦ correct number of instances ♦ adding the correct associations	Entities could be written in reverse order	3	Analysis
			
2b	1 mark for each bullet. ♦ fields, function, alias - forename, surname, plannerNo - COUNT of records with alias ♦ equijoins ♦ GROUP BY (forename), (surname), plannerNo ♦ ORDER BY COUNT of records DESC	SELECT Planner.forename, Planner.surname, Planner.plannerNo, COUNT(Walk.walkID) AS [Total Participants] FROM Walk, Route, Planner WHERE Walk.routeID = Route.routeID AND Route.plannerNo = Planner.plannerNo GROUP BY Planner.forename, Planner.surname, Planner.plannerNo ORDER BY COUNT(Walk.walkID) DESC; Count can be on any field or * SQL allows ordering on alias for Count. 4 records produced	4	Implementation

Task	Expected response	Additional guidance	Marks available	
2	**Database design and development**			
2c	1 mark for each bullet. ♦ query to find longest walk using Max function ♦ correct fields and tables ♦ equi-joins ♦ use of Query 1 result in the criteria to produce correct output ♦ use of GROUP BY walkerNo (forename, surname, telNo) to remove walkers who have walked the longest route more than once	Query using MAX function could be a sub-query within the criteria. SELECT MAX(distance) AS longest FROM Route; (query saved as LongestWalk) SELECT Walker.walkerNo, Walker.forename, Walker.surname, Walker.telNo FROM Walker, Walk, Route, LongestWalk WHERE Walk.routeID = Route.routeID AND Walker.walkerNo = Walk.walkerNo AND distance = longest GROUP BY Walker.walkerNo, Walker.forename, Walker.surname, Walker.telNo; Award 0 marks for bullet 3 if innerjoin is used Award 0 marks for bullet 4 if value is used instead of field.	5	Implementation
2d	1 mark for each bullet. ♦ use of wildcard before shoe/s ♦ printed evidence of new SQL statement accommodating additional types of shoe producing same output		2	Testing
2e	♦ find the travelling distance from a walker's home to the starting point of a chosen route OR ♦ display walkers who prefer to walk a route with a chosen level of difficulty		1	Evaluation

Task	Expected response	Additional guidance	Marks available	
3	**Web design and development**			
3a	End user 1 mark for any bullet. Max 2 marks. ♦ read/find/get information about woodland walks ♦ read/buy maps (with route directions) ♦ read/get information about local wildlife ♦ read/get information about lengths of walks ♦ read/get information about the rating of walks ♦ book walking tours online Functional 1 mark for any bullet. Max 1 mark. ♦ order form/link to buy maps. ♦ form/link to book walking tours online. ♦ navigation bar to move between pages of the website ♦ formatted web pages displaying text and graphics (in context)	Must indicate a feature of the software.	3	Analysis
3b	1 mark for each bullet. ♦ (annotated) wireframe showing walks, description and map ♦ description of show/reveal when walk element is selected	There may be a variety of approaches here but any solution should limit info to a page but still retain a method by which an element will show/reveal walks individually	2	Design

Task	Expected response	Additional guidance	Marks available	
3	**Web design and development**			
3c	1 mark for each bullet. ♦ IDs assigned to sections (walks) ♦ Walk 2 and Walk 3 sections set to display:none ; ♦ onmouseover events added to h3 elements displayed on one line ♦ functions displaying one element while hiding the others		4	Implementation
3d	1 mark for each bullet. ♦ creating sections/divs for each resized image and description ♦ use of float to position graphics and text to right or left as per design ♦ addition of Gallery link to nav bar	Accept various implementations of float attribute Gallery link not required on other pages.	3	Implementation
3ei	1 mark for each bullet. ♦ test should include no selection AND multiple selections of a wood ♦ test should include inputs within a range 0 to 5 AND outwith range 0 to 5	Exceptional input should not be text as the web page does not allow this to be entered.	2	Testing
3eii	1 mark for any bullet. Max 1 mark. ♦ lack of validation on text fields (invalid contact emails etc) ♦ form can be submitted without a wood selected ♦ user cannot book multiple walks on the same form		1	Evaluation

Task 1 – Software Design and Development		Marks Available	Marks Awarded
1a – Analysis	Assumptions	2	
			/2
1b – Design	Identify first character	1	
	Upper-case conversion	1	
	Concatenation	1	
	String/variable returned	1	
			/4
1c - Implementation			
Read in mammals data	Module with parameter	1	
	Sightings data stored in record structure and assigned to an array	1	
Find oldest person	Module with parameter	1	
	Initialise and re-assign max age	1	
	IF statement	1	
Upper-case function	Extract first character	1	
	IF statement	1	
	Return original or concatenated string	1	
Display dates of sightings	Module with parameter	1	
	Linear search	1	
Display number of sightings each day	Module with parameter	1	
	Count initialised and incremented	1	
	dayToCount and count reset	1	
Implementation	Single upper-case function called twice	1	
	Modular and maintainable	1	
			/15
1d – Testing	Watchpoint set	1	
	Counted correctly	1	
			/2
1e – Evaluation	Efficiency of function	1	
	Maintainability – modularity	1	
			/2

Task 2 – Database Design and Development		Marks Available	Marks Awarded
2a – Design	Entities	1	
	Instances	1	
	Associations	1	
			/3
2b– Implementation	Fields, function and alias	1	
	Equi joins	1	
	Group by	1	
	Order by	1	
			/4
2c – Implementation	Query using Max function	1	
	Fields and tables	1	
	Equi-joins	1	
	Use of Query 1	1	
	Group by	1	
			/5
2d – Testing	Use of wildcard	1	
	Alias/printout	1	
			/2
2e – Evaluation	Functional requirement	1	/1

Task 3 – Web Design and Development		Marks Available	Marks Awarded
3a – Analysis	End-user requirements	2	
	Functional requirement	1	
			/3
3b – Design	Wireframe	1	
	Description of show/reveal	1	
			/2
3c – Implementation	IDs assigned to sections	1	
	Walks 2 & 3 sections display none	1	
	Onmouseover events on h3 element	1	
	Function(s)	1	
			/4
3d – Implementation	Sections/divs	1	
	Float	1	
	Gallery link	1	
			/3
3e(i)– Testing	None and multiple selections	1	
	Range of numbers	1	
			/2
3e(ii) – Evaluation	Form evaluation	1	/1

[END OF MARKING INSTRUCTIONS]