| Task | Expected response | Additional guidance | Marks available | |
|---|---|---|---|---|
| | **Software design and development** | | | |
| 1c(i) | Read in attraction data (2)<br>♦ module with correct parameters passed or returned<br>♦ assigned to five parallel arrays<br><br>Find least and most visited (4)<br>♦ module with attraction() and visitors() passed in<br>♦ initialise minimum and maximum value or position<br>♦ find minimum and maximum value or position<br>♦ traverse array to display the names of the least visited (Beaver Falls, CandyFloss Carousel) and most visited (Thundergun Express) attractions<br><br>Write to file (4)<br>♦ module with attraction() and category() and daysOpen() passed in<br>♦ create/open, write to and close service.csv file<br>♦ use of modulus to calculate the number of days until next service<br>♦ two if conditions met<br>○ = rollercoaster<br>○ <=7<br><br>Implementation (2)<br>♦ modular program matches top-level design (three sub-programs) and refinements<br>♦ program is maintainable | If candidate passes in array length variable in each sub-routine award 0 marks for first instance.<br><br>If candidate uses array of records award 0 marks for Read procedure then accept appropriate parameter passing for remaining procedures/ functions.<br><br>Do not penalise separate find min/max subroutines with attraction() and visitors() at this point.<br><br>Award bullet point 3 (loop, If, assignment correct) even if bullet point 2 is incorrect and output may be wrong.<br><br>Award 0 marks for bullet point 3 if pre-defined function is used.<br><br>(Asteroid Belt, G-Force, Sonic Boom, Vortex)<br><br>Do not penalise if additional functions are called from within step two (find min/max).<br>Meaningful variable/sub-program names and internal commentary relevant to task. | 15 | Implementation |

<table>
<tr><th>Task</th><th>Expected response</th><th>Additional guidance</th><th>Marks available</th><th></th></tr>
<tr><td></td><td colspan="2">Software design and development</td><td></td><td></td></tr>
<tr><td>1c(ii)</td><td>Award 1 mark for each bullet:<br><br>Count and display (3)<br>♦ module with height() passed in<br>♦ identify height of 1m or over<br>♦ initalise and increment count(16)</td><td></td><td>3</td><td></td></tr>
<tr><td rowspan="2">1d</td><td>Award 1 mark for each correct column</td><td>Accept other expression of true (yes/1) and false (no/0)<br><br>Accept repetition of 65 and false for second iteration</td><td rowspan="2">3</td><td rowspan="2">Testing</td></tr>
<tr><td>Sample answer
<table>
<tr><th></th><th>If current category is ‘Roller Coaster’</th><th>days</th><th>If (90 – days) is less than or equal to 7</th></tr>
<tr><td>1st iteration</td><td>true</td><td>65</td><td>false</td></tr>
<tr><td>2nd iteration</td><td>false</td><td></td><td></td></tr>
<tr><td>3rd iteration</td><td>true</td><td>83</td><td>true</td></tr>
</table></td></tr>
</table>

| Task | Expected response | Additional guidance | Marks available | |
|---|---|---|---|---|
| | **Software design and development** | | | |
| 1e | Award 1 mark for bullet point from data structure.<br>Award 1 mark for bullet point from loops.<br>Max 2 marks.<br><br>Data Structures<br>♦ array size is fixed (26 elements) and would need to be updated.<br>OR<br>♦ the array is initialised with no size and new elements are appended/arrays are redimensioned<br><br>Loops<br>♦ number of iterations of loops are fixed (26)<br>OR<br>♦ the loop iterates to the length of the array/end of file | Candidate response must correspond with their code. | 2 | Evaluation |