|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | For Examiner’s Use | | | Examiner’s Initials | | | Question | Mark | | **1** | 1 | | **2** | 9 | | **3** | 4 | | **4** | 4 | | **5** | 13 | | **6** | 1 | | **7** | 8 | | **8** | 3 | | **9** | 5 | | **10** |  | | **11** |  | | **12** |  | | **13** |  | | **TOTAL** |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Centre number** |  |  |  |  |  | **Candidate number** | 0 | 2 | 2 | 6 | | **Surname** | DEAL | | | | | | | | | | | **Forename(s)** | JAMES | | | | | | | | | | | **Candidate signature** |  | | | | | | | | | | | **Programming Language** | **PYTHON (7517/1D)** | | | | | | | | | |   **A-LEVEL COMPUTER SCIENCE** |
| **Paper 1** |

Monday 3 June 2019

Morning

Time allowed: 2 hours 30 minutes

Instructions

* This is the Electronic Answer Document (EAD). Answer **all** questions by entering your answers into this document on screen. You **must save** this document at regular intervals.
* Before the examination begins, type the information needed in the boxes **at the top of this page**.
* Before the examination begins, type the information needed in the boxes **in the footers** (page 2 onwards) of this EAD.

**During the examination**

* You may print pages of your EAD. A print monitor will collect and deliver your print-out to you. You must **not** collect your own print-out.

**Exceptions**

* If you experience difficulty inserting screen shots into your EAD then you may print these separately and attach them to the back of the EAD with a reference in the correct place in the EAD. Ensure that your **Centre Number**, **Candidate Name** and **Candidate Number** are on each sheet.

**At the end of the examination**

* Save for the last time and print your EAD **on one side only** (not double-sided). A print monitor will collect and deliver your print-out to you. Check that your details are in the footers of every page. Write them in if they are not.
* Enter your signature on the front cover.
* Staple or tie all pages together in the top left-hand corner of the EAD.
* Hand in **all** pages of the EAD to the Invigilator.

**Warning**

* No extra time is allowed for printing and collating.
* It may not be possible to credit an answer if your:

– details are not printed on every page as instructed above

– screen captures are not legible to the Examiner.

Answer **all** questions.

You **must save** this document at regular intervals or you may lose your work.

**Section A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 01** | | | | |  |  |
| **0** | **1** | **.** | **1** | Firstly you could remove the triple assignment in favor of tuple assignment which would allow the assignment to be executed without a temp value.  - Have a flag variable that is set to True if a swap is made and result to False at the start of each pass.  - Change the outer loop so that it would stop if no swaps have been made.  - After inner loop subtract 1 from N. | | 0/4 |
| **0** | **1** | **.** | **2** | As some sorting algorithms have very efficient big O complexities which make adding more elements to the list plateau in added complexity as per a O(logn) complexity.  - Sorting a list is always a tractable problem.  - As the size of the problem grows it does not affect whether it is tractable or intractable. | | **0/2** |
| **0** | **1** | **.** | **3** | A programmer tasked with solving an intractable problem may engage in proceddural decomposition to break down the larger problem into smaller more manageable chunks.  - An algorithm that makes an estimate based on experience.  - A solution that only works in some cases.  - Relax some contraints of the problem. | | **1/2** |
|  |  |  |  |  | | **1** |
|  |  |  |  |  | | **8** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | *Leave blank* |
| **Question 02** | | | | |  | |  |
| **0** | **2** | **.** | **1** | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 0 | # | 1\* | 0 | 1 | 0 | 0 | | 0 | #\* | 1 | 0 | 1 | 0 | 0 | | 0 | # | 1\* | 0 | 1 | 0 | 0 | | 0 | # | # | 0\* | 1 | 0 | 0 | | 0 | # | # | 0 | 1\* | 0 | 0 | | 0 | # | # | 0 | 1 | 0\* | 0 | | 0 | # | # | 0 | 1\* | 1 | 0 | | 0 | # | # | 0\* | 1 | 1 | 0 | | 0 | # | #\* | 0 | 1 | 1 | 0 | | 0 | # | # | 0\* | 1 | 1 | 0 | | 0 | # | #\* | 0 | 1 | 1 | 0 | | 0 | #\* | 1 | 0 | 1 | 1 | 0 | | 0\* | 1 | 1 | 0 | 1 | 1 | 0 | | 0 | 1\* | 1 | 0 | 1 | 1 | 0 | | | |  | | --- | | S4 | | S4 | | S0 | | S1 | | S2 | | S2 | | S3 | | S3 | | S4 | | S0 | | S5 | | S5 | | S5 | | S6 | | 5/5 |
| **0** | **2** | **.** | **2** | The purpose of the Turing machine is to copy a string of 1’s | | | 1/1 |
| **0** | **2** | **.** | **3** | To place the read/write head onto the start of the original string and to set an accepting state. | | | 1/1 |
| **0** | **2** | **.** | **4** | A universal turing machine is a turing machine which can emulate a turing machine within itself. It also has an infinite ticker tape which simulates the inputs and outputs of the emulated turing machine. | | | 2/2 |
|  |  |  |  |  | | | 9 |
|  |  |  |  |  | | | **9** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 03** | | | | |  |  |
| **0** | **3** | **.** | **1** | |  |  |  | | --- | --- | --- | | I | E | H | | C | A | B | | G | D | F | | | 1/1 |
| **0** | **3** | **.** | **2** | Reprsentational abstraction is representing the problem in a graphical way to break it down into smaller steps.  - Removing unecessary details. | | 0/1 |
| **0** | **3** | **.** | **3** | Abstraction by generalisation is where the problem is generalised to more situations. Making it more broad and simpler to solve.  - Grouping by common characteristics. | | 0/1 |
| **0** | **3** | **.** | **4** | The borders of each quadrant have been removed. | | 1/1 |
| **0** | **3** | **.** | **5** | An adjancey matrix is more appropriate when one node has many connections to other nodes. When the the graph is not sparse and connections are often changed. | | 2/2 |
| **0** | **3** | **.** | **6** | Node graph  - Directed graph. | | 0/1 |
|  |  |  |  |  | | 4 |
|  |  |  |  |  | | **7** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 04** | | | | |  |  |
| **0** | **4** | **.** | **1** | The \* metacharacter means 0 or more. | | 1/1 |
| **0** | **4** | **.** | **2** | The ? Metacharacter means 0 or 1. | | 1/1 |
| **0** | **4** | **.** | **3** | |  | | --- | | Y | | N | | Y | | N - Y | | N | | N | | N | | | 2/3 |
|  |  |  |  |  | | 4 |
|  |  |  |  |  | | **5** |

**Section B**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 05** | | | | |  |  |
| **0** | **5** | **.** | **1** |  | | 12/12 |
| **0** | **5** | **.** | **2** |  | | 1/1 |
|  |  |  |  |  | | 13 |
|  |  |  |  |  | | **13** |

**Section C**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 06** | | | | |  |  |
| **0** | **6** | **.** | **0** | As it is a universal datatype that all computers can directly read. It allows binary operations to be performed on data directly.  - Binary file cannot be easily read which means game data is obfuscated from users.  - File size likely to be smaller. | | 1/2 |
|  |  |  |  |  | | 1 |
|  |  |  |  |  | | **2** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 07** | | | | |  |  |
| **0** | **7** | **.** | **1** | The item is not found. | | 1/1 |
| **0** | **7** | **.** | **2** | It checks both against the name and ID parameter but accepts if either is True using an OR.  - It searches by name if the ID parameter has a value of -1. | | 0/1 |
| **0** | **7** | **.** | **3** | O(n) complexity | | 1/1 |
| **0** | **7** | **.** | **4** | If two items have the same name it will always return the first one’s index even if meant to get the second one. | | 2/2 |
| **0** | **7** | **.** | **5** | In order to find an item in a hash table, the hashing algorithm is first applied to the item. This then provides a location in the hashing table. If we check this location and it is not the item we’re searching for, we can move onto the next location as it’s likely there was an assignment collision and the item is in the next slot. We can then return the data for that hash.  - If the location is empty then the item does not exist. | | 3/4 |
| **0** | **7** | **.** | **6** | As the number of items is relatively small and therefore the computation for a hash table is unecessary. | | 1/1 |
|  |  |  |  |  | | 8 |
|  |  |  |  |  | | **10** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 08** | | | | |  |  |
| **0** | **8** | **.** | **1** | To work out the set of items the player can use in the current game state we can perform an intersection between set B as it has all the items that have “use” in their commands and set D unioned with set E as it contains all the items in the players inventory and all the items which are in players current location and have “usable” ststaus respectively. An intersection between these two sets will provide the items that the player can use at this point. | | 2/3 |
| **0** | **8** | **.** | **2** | The set A, The set B | | 1/1 |
| **0** | **8** | **.** | **3** | As not all items which are “getable” will be in the same location as the player.  - because there could be items in a cotainer object that is in the current location. | | 0/1 |
| **0** | **8** | **.** | **4** | A proper subset has all of items within the parent set. Not just some.  - There will be at least one value in a set that is not in the proper subset of that set, this does not have to be the case for a subset. | | 0/1 |
|  |  |  |  |  | | 3 |
|  |  |  |  |  | | **6** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 09** | | | | |  |  |
| **0** | **9** | **.** | **1** | main | | 1/1 |
| **0** | **9** | **.** | **2** | PlayGame | | 1/1 |
| **0** | **9** | **.** | **3** | LoadGame() | | 1/1 |
| **0** | **9** | **.** | **4** | A local variable is only accesible within a particular part of the program whereas a global variable is accessible by all parts. | | 1/1 |
| **0** | **9** | **.** | **5** | As it can help to prevent side effects where variables in other parts are changed unintentionally. | | 1/1 |
|  |  |  |  |  | | 5 |
|  |  |  |  |  | | **5** |

**Section D**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 10** | | | | |  |  |
| **1** | **0** | **.** | **1** |  | |  |
| **1** | **0** | **.** | **2** |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | | **5** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 11** | | | | |  |  |
| **1** | **1** | **.** | **1** |  | |  |
| **1** | **1** | **.** | **2** |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | | **8** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 12** | | | | |  |  |
| **1** | **2** | **.** | **1** |  | |  |
| **1** | **2** | **.** | **2** |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | | **13** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | *Leave blank* |
| **Question 13** | | | | |  |  |
| **1** | **3** | **.** | **1** |  | |  |
| **1** | **3** | **.** | **2** |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | | **9** |

**If you have made any separate screen captures, put a reference in the correct place in this Electronic Answer Document.**

**Ensure that your Centre Number, Candidate Name and Candidate Number are on each sheet.**

**Attach them to the back of this document.**

NB: The AQA logo is provided for use only in the Electronic Answer Document for the 2019 A-level Computer Science Paper 1 (7517/1) examination and must not be used for any other purpose.

Copyright © 2019 AQA and its licensors. All rights reserved.