|  |  |
| --- | --- |
| Y:\Operations\Document_Production\Documents\Mark Schemes\Resources\logos\IGCSE.eps | **Cambridge Assessment International Education** Cambridge International General Certificate of Secondary Education |

Computer SCIENCE **0478/22**

Paper 2   **June 2021**

MARK SCHEME

Maximum Mark: 50

**IMPORTANT NOTICE**

Mark Schemes have been issued on the basis of **one** copy per Assistant examiner and **two** copies per Team

Leader.

|  |
| --- |
| **Pre-Standardisation** |

**Generic Marking Principles**

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

|  |
| --- |
| GENERIC MARKING PRINCIPLE 1:  Marks must be awarded in line with:   * the specific content of the mark scheme or the generic level descriptors for the question * the specific skills defined in the mark scheme or in the generic level descriptors for the question * the standard of response required by a candidate as exemplified by the standardisation scripts. |
| GENERIC MARKING PRINCIPLE 2:  Marks awarded are always **whole marks** (not half marks, or other fractions). |
| GENERIC MARKING PRINCIPLE 3:  Marks must be awarded **positively**:   * marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate * marks are awarded when candidates clearly demonstrate what they know and can do * marks are not deducted for errors * marks are not deducted for omissions * answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous. |
| GENERIC MARKING PRINCIPLE 4:  Rules must be applied consistently e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors. |
| GENERIC MARKING PRINCIPLE 5:  Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen). |
| GENERIC MARKING PRINCIPLE 6:  Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind. |

**Please note the following further points:**

Please also read the additional guidance in the mark scheme, it provides further information about how to mark the question.

Please annotate **every** question on your scripts. The number of ticks given **must** match the number of marks given. If you award a benefit of doubt (BOD) mark, this **must** also have a **tick**.

The sections in brackets in the mark scheme are not necessary in the candidates answer.

The words in **bold** in the mark scheme are important text that needs to be present, or some notion of it needs to be present. It doesn’t have to be the exact word, but something close to the meaning.

If a word is underlined, this **exact** word must be present.

A single forward slash means this is an alternative word. A double forward slash means that this is an alternative mark point.

Red text in the mark scheme is a response that we think it just about okay, but that will not be published as a response.

Ellipsis (…) on the end of one mark point and the start of the next means that the candidate **cannot** get the second mark point without being awarded the first one. If a MP has ellipsis at the beginning, but there are no ellipsis on the MP before it, then this is just a follow-on sentence and **can** be awarded **without** the previous mark point.

Please inform your team leader when you have submitted your standardisation scripts. They may have many examiners that they are monitoring, so may not see your submission immediately.

The mark scheme has been agreed at standardisation, so it is the AE’s responsibility to apply it. You may have differing opinions on how the paper should be marked, but the mark scheme is what has been agreed by a panel and AE’s are required to apply it.

Please mark your allocation at a steady rate. If you are not able to mark your allocation for a number of days, inform your team leader, do not wait for them to have to chase you. Please ensure that you meet the 40% deadline. If you are not close to or meeting this at the 40% deadline, some of your allocation may be reallocated. If you are going to struggle with this deadline, you **must** inform your team leader ASAP.

There is a blank page(s) in the exam paper. This **must** be annotated with a **SEEN** annotation, to indicate it has been checked for any further responses. Also, any blank responses must have a SEEN annotation, as every question must be annotated. If you do not annotate correctly, you may be stopped by your TL.

If a candidate writes outside the zoned area for the question, this must be linked to the response, even if it is not awarded a mark. This demonstrates at grade review that you did read this part of the response.

| **Question** | **Answer** | **Marks** | **Guidance** |
| --- | --- | --- | --- |
|  | **Section A** |  |  |
| 1 (a) | Array TrainUp  Data type Integer/int  Use Storing the number of passengers on the train journeys up  Many correct answers, they must be meaningful. The names are examples only. | **3** |  |
| 1(b) | Validation check (type check) to check that the number entered is a whole number  Normal data 34  Erroneous data two  Validation check (range check) to check that the value of the number entered is between 1 and 80 inclusive  Normal data 34  Erroneous data 99  Many correct answers, the data are examples only. | **6** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1(c) | Any **six** from:  Input number of tickets  Input train up mountain and train down mountain  Suitable prompts for all inputs included  Check that tickets are available  If available calculate price of tickets …  … calculation includes discount if required  Update total passengers for correct up train  Update total passengers for correct down train  Total cost up and down    Sample  OUTPUT “How many tickets”  INPUT NoTickets  OUTPUT “Which Train up the mountain? 1, 2, 3 or 4”  INPUT UpNumber  OUTPUT “Which Train down the mountain? 1, 2, 3 or 4”  INPUT DownNumber  IF (DownNumber = 4 AND TrainUp[UpNumber] + NoTickets <= 480 AND TrainDown[DownNumber] + NoTickets <= 640) OR (TrainUp[UpNumber] + NoTickets <= 480 AND TrainDown[DownNumber] + NoTickets <= 480)  THEN  Cost ← (NoTickets – NoTickets DIV 10) \* 50  TrainUp[UpNumber] ← TrainUp[UpNumber] + NoTickets  TrainDown[DownNumber] ← TrainUp[DownNumber] + NoTickets  TrainUpTotal[UpNumber] ← TrainUpTotal[UpNumber] + Cost / 2  TrainDownTotal[DownNumber] ← TrainDownTotal[DownNumber] + Cost / 2  ENDIF | | **6** |  |
| 1(d) | | Explanation  Any **five** from:  How the program displayed the number of passengers for each journey with a suitable message  How the program displayed the amount of money taken for each journey with a suitable message  How the program calculated and displayed the total number of passengers with a suitable message  How the program calculated and displayed the total money taken with a suitable message  How the program selected the train journey with the most passengers  How the program dealt with more than one train being the most popular  How the program displayed the most popular train journey with a suitable message  Programming statements should be used and must be explained. | **5** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Section B** | |  |  |
| 2 (a) | Any **six** from:  Initialisation of large and small variables  Use of loop for 500 entries  Input  Checking the range  Checking for a whole number  Selecting largest number  Selecting smallest number  Outputting the largest, smallest and range  Large ← 0  Small ← 1000  FOR Count ← 1 TO 500  REPEAT  OUTPUT “Enter a whole number between 1 and 999”  INPUT Number  UNTIL Number >= 1 AND Number < 1000 AND Number = Number DIV 1  IF Number < Small  THEN  Small ← Number  ENDIF  IF Number > Large  THEN  Large ← Number  ENDIF  NEXT  Range ← Large – Small  OUTPUT “Largest number is “, Large  OUTPUT “Smallest number is “, Small  OUTPUT “Range of numbers is “, Range | | **6** | * Review at STM. Should the issue of whether zero is included as a positive number. The sample algorithm in the mark scheme excludes zero, I suggest that its inclusion may need to be condoned. |
| 2(b) | | Reduce the amount of numbers entered  By decreasing the final value of the loop  or  Remove the need to input values  By using random numbers / a previously populated array | **2** | Review at STM. The instructions to examiners probably need to be a bit more explicit. Is it any two mark points from the four listed, or one mark for a change and a further mark for an expansion? |
| 3(a) | **Pseudocode statement Flowchart symbol**  IF Number = 20  PRINT Number  Number ← Number + 1 | | **3** | Allow  For PRINT |
| 3(b) | IF Number = 20 selection  PRINT Number output  Number ← Number + 1 counting | | **3** | Consider assignment instead of counting at STM |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4(a) | One mark for each correct column   |  |  |  | | --- | --- | --- | | **Password** | **PasswordRepeat** | **OUTPUT** | |  |  | Please enter password | | Secret |  | Reject | |  |  | Please enter password | | Secret |  | Reject | |  |  | Please enter password | | VerySecret |  |  | |  | VerySecret | Accept | |  |  | Please enter password | | Pa55word |  |  | |  | Pa55word | Accept | |  |  | Please enter password | | 999 |  | Reject | | **3** |  |
| 4(b) | Any **four** from:  Include INPUT PasswordRepeat in suitable loop // repeat input  Allow up to three tries  Use of appropriate messages  Exit the loop after three tries or if the repeated password matches the original  On exit output Reject if all tries are incorrect  On exit output Accept if the entry is correct | **4** | Review at STM. Do the last two mark points answer the question? |

|  |  |  |  |
| --- | --- | --- | --- |
| 5(a) | **One** mark for:  Use of FOR loop  Correct number of Iterations  Assignment  FOR Count ← 1 TO 20  dataArray[Count] ← 0  NEXT (Count) | **3** |  |
| 5(b) | A FOR loop has a fixed number of repetitions // No need to manage the loop counter | **1** | Allow FT |
| 6 | **One** mark for:  TABLE row not completed  POSITION column not required// POSITION criteria not required  No criteria set in the size and flower columns   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Field: | SIZE | PRICE | FLOWER | NUMBERSOLD | NAME | | Table: | PLANT | PLANT | PLANT | PLANT | PLANT | | Sort: |  |  |  |  |  | | Show: | 🞏 | 🗹 | 🞏 | 🗹 | 🗹 | | Criteria: | ="small" |  | False |  |  | | or: |  |  |  |  |  |   **One** mark for correct rows Field, Table and Show  **One** mark for correct Criteria row | **5** | Review at STM. The instructions to examiners could be a bit more explicit. |