**TERM 1 REVISION EXAM MARKING SCHEME**

**YEAR 9 - CS - Paper 2**

|  |  |  |
| --- | --- | --- |
| **Question** | **Answer** | **Marks** |
| 1(a) | **One mark per mark point, max six**   * Line 1 100 * Line 7 Value > 100 // Value >= 101 * Line 11 Reading[Value] + 1 * Line 14 INPUT Value * Line 18 Reading[Count] * Line 19 Count – 1 | 6 |
| 1(b) | One mark per mark point, max three   * use an IF/conditional statement * to check if Reading[Count] not equal to zero * before outputting the value // between statements 17 and 18 // code sample showing position   IF Reading[Count] <> 0  THEN  OUTPUT  ENDIF | 3 |
| 2 | One mark per row, max four | 4 |
| 3 | One mark per mark point, max four  - variables are used to represent values that can change during the execution of a program // variables can be used to store the results of calculations / counting / totaling // can store values entered by the user  - variable example – any data that is input into a program such as a date  -constants represent values that must stay the same throughout the execution of a program  -constant example – any value that does not change, such as Pi in mathematical formulae | 4 |
| 4(a) | One mark per mark point, max seven  MP1 correct In column  MP2 correct Logic column  MP3 correct Test column  MP4 correct Number column  MP5 correct Store[Count] column  MP6 correct Count and Limit columns  MP7 correct Out and OUTPUT columns | 7 |
| 4(b) | One mark per mark point, max two  - to find / output prime numbers  - ... store prime numbers in an array | 2 |
| 4(c) | One mark per mark point, max three  MP1 insert a WHILE loop ... // pre-condition loop  MP2 ... after Input Number  MP3 ... with a condition to enter the loop Number < 3  MP4 an error message included within the loop to ask for a re-entry of Number  MP5 ...with another input prompt for Number  MP6 ENDWHILE closes the loop and the program carries on from REPEAT in the original algorithm  OR  One mark per mark point, max three  MP1 insert a REPEAT loop ... // post-condition loop  MP2 ... before Input Number  MP3 a conditional statement should be placed after Input Number  MP4 ...to check if Number < 3  MP5 if the number entered is <3, an error message included within the loop to ask for a re-entry of Number  MP6 UNTIL Number >= 3 closes the loop and the program carries on from REPEAT in the original algorithm. | 3 |
| 5(a) | One mark per mark point, max three   * line 8 / PassCheck<- TRUE   correction PassCheck<- FALSE   * line 12 / IF Password <> Password   correction IF Password2 <> Password // IF Password <> Password2   * line 18 / UNTIL PassCheck OR Attempt <> 3   correction UNTIL PassCheck OR Attempt = 3 / UNTIL PassCheck OR Attempt >= 3 | 3 |
| 5(b) | One mark check, one mark matching description, max four  Check: validation // length check  Description length check // checks number of characters in password  Check: verification // double entry  Description double entry // comparison that two inputs are the same | 4 |
| 5(c) | One mark per set, one mark matching reason, max four  Set 1 – any appropriate example e.g. “small”  Reason: must follow through from the password given e.g. abnormal data will be rejected  Set 2 – any different appropriate example e.g. “password” and “password”  Reason: must be different and follow through from the password given e.g. normal data will be accepted | 4 |