**TERM 1 Revision MARKING SCHEME**

**YEAR 8 - CS**

|  |  |  |
| --- | --- | --- |
| 1(a) i |  Sound | 1 |
| 1 a (ii) | Lossy compressed file | 1 |
| 1 b | Computer consist of transistors / logic circuits/gates … … that can only store/process data in two states / high-low / on-off / 1 and 0 | 2 |
| 1 c | 01000000 01100101 11110010 | 3 |
| 1 d | 0100 (1 mark) 0010 (1 mark) 1100 (1 mark) 1110 (1 mark) | 4 |
| 1 e | Two marks for two correct stages of working, one mark for correct final answer • 100 × 150 • 15 000 × 16 // 15 000 × 2 • 240 000 / 8 • 30 000 bytes | 3 |
| 1 f | One mark for full method of working e.g. conversion to binary then flipping and adding 1 One mark for correct answer • 10110010 | 2 |
| 1 g |  | 3 |
| 1 h | Two from: • The result of the calculation is greater than 255 // The value generated is larger than can be stored in the register • The result of the calculation would require more than 8 bits to be represented // A register has a predetermined number of bits and there are too many bits for it | 2 |
| 2 a | One mark for each correct definition: • The sample rate is the number of samples taken in a second/per time unit • The sample resolution is the number of bits per sample | 2 |
| 2 b | Lossy compression | 1 |
| 2 c (i) | Any two from: e.g. • Destination/receivers (IP) address • Packet number • Originator’s/senders (IP) address | 2 |
| 2 c (ii) | Any five from: • Data is broken/split/divided into packets • Each packet (could) take a different route • A router controls the route/path a packet takes • … selecting the shortest/fastest available route/path • Packets may arrive out of order • Once the last packet has arrived, packets are reordered • If a packet is missing/corrupted, it is requested again | 5 |
| 3a | One mark per each correct character in the correct order: • 9 • 3 • 0 • D | 4 |
| 3b(i) | 00001111 | 1 |
| 3b(ii) | Any one from: • The value becomes incorrect/inaccurate as the right most bits are lost • It is divided by 8 | 1 |
| 3c | Any two from: • Easier/quicker to understand/read/write • Easier/quicker to debug • Less likely to make a mistake • Shorter representation // Takes up less screen space | 2 |
| 3d | One mark for two correct characters, two marks for three correct characters in the correct order: • 1 • 2 • D | 2 |
| 4 a | Any three from: • A character set is used • … such as Unicode/ASCII • Each character has a unique binary value | 3 |
| 4 b (i) | It reduces the file size | 1 |
| 4 b (ii) | Any four from: • A compression algorithm is used • … such as RLE/run length encoding • Repeating words/characters/phrases are identified // Patterns are identified • … and indexed • … with number of occurrences • … with their position | 4 |
| 4 b (iii) | Any two from: e.g. • To save storage space • To make it quicker to transmit • To make it small enough to attach to an email • To reduce the bandwidth needed to transmit | 2 |
| 5  |  | 5 |
| 6 a |

|  |  |
| --- | --- |
| **Initializing Total Variable** | 1 |

|  |  |
| --- | --- |
| **FOR Loop Structure** | 1 |

|  |  |
| --- | --- |
| **Input within Loop** | 1 |

|  |  |
| --- | --- |
| **Accumulating Total** | 1 |

|  |
| --- |
| **Outputting the Result 1** |

pseudocodetotal <- 0FOR i <- 1 TO 10  OUTPUT "Enter a number: ", number total <- total + numberNEXT iOUTPUT "The total is: ", totalPython codetotal = 0for i in range(10): number = int(input("Enter a number: ")) total = total + numberprint("The total is:", total) | 5 |
| 6 b | 1 mark for each correct circle | 5 |
| 7 | One mark per mark pointMP1 Correct input statement with appropriate variableMP2 Elements of selection statement present – CASE OF ENDCASEMP3 At least one correct branch in the case statementMP4 All branches from 1 to 3 correctMP5 Correct use of OTHERWISE with correct output.For example:INPUT NumberCASE OF Number 1 : OUTPUT Number 2 : OUTPUT Number 3 : OUTPUT Number OTHERWISE OUTPUT "ERROR"ENDCASEOrINPUT NumberCASE OF Number 1 : OUTPUT 1 2 : OUTPUT 2 3 : OUTPUT 3 OTHERWISE OUTPUT "ERROR"ENDCASE | 5 |
| 8 | 1 mark for CASE OF1 mark for correct case statement1 mark for OTHERWISE1 mark for endcaseCASE OF variable CASE 1: statement CASE 2: statement OTHERWISE statementENDCASE | 4 |