1(a)(i)	B3A
1(a)(ii)	0001 0000 1000
1(a)(iii)	**1 mark** for the working **1 mark** for the correct denary value **Working:** Method 1: 1111 1011 1100 Flip the bits 0000 0100 0011 Add 1 0000 0000 0001 + 0000 0100 0100 Method 2: -2048 + 1024 + 512 + 256 + 128 + 32 + 16 + 8 + 4 // -128 + 32 + 16 + 8 + 4 **Denary value:** -68
1(b)(i)	(1) 00101011
1(b)(ii)	**1 mark** per bullet point, **max 2 marks** • An overflow error occurs • The answer cannot be represented in the number of bits available // The answer is larger than the maximum positive number that can be stored in the register // The answer is smaller than the most negative number that can be stored in the register

2(a) **3 marks** for 6 correct ticks
2 mark for 4 or 5 correct ticks
1 mark for 2 or 3 correct ticks

Example of data	Number of bits						
	4	**8**	**16**	**24**	**32**	**64**	**128**
the hexadecimal value F139			✓				
16 000 000 unique amplitude values				✓			
an IPv4 address					✓		
256 unique colours		✓					
an IPv6 address							✓
the denary value 65 000			✓				

2(b) **1 mark**

1111 1001 0100

2(c) **1 mark**

1111 0010

2(d) **1 mark**

E33B

2(a)	**1 mark** for: 0010 0010 1110 **1 mark** for: 22E
2(b)(i)	**1 mark** for: -30
2(b)(ii)	**1 mark** for smallest, **1 mark** for largest Smallest: 1000 0000 Largest: 0111 1111
2(c)	**1 mark** for application and **1 mark** for corresponding justification e.g. • an application that performs financial / banking calculations • because financial transactions use only two decimal places and must be accurate, no accumulating / rounding errors and it is difficult to represent decimal values exactly in normal binary. Or • electronic displays • because visual displays only need to show individual digits • because conversion between denary and BCD is straightforward Or • The storage of the date and time in the BIOS of a PC • because conversion between denary and BCD is more straightforward Or • Barcode systems • because conversion between denary and BCD can be accurately completed

6(a)	**1 mark** for identification **1 mark** for matching description e.g. • ASCII • 7/8 bits per character // represents 128/256 characters // represents all characters from Latin alphabet • UNICODE • 8/16/32 bits per character // represents 256/65536+ characters // represents all characters in all languages
6(b)(i)	**1 mark** for: 256 // 2^8
6(b)(ii)	**1 mark** for: Increased file size
6(b)(iii)	**1 mark** for each bullet point (**max 2**) e.g. • The change may not be noticeable // Data removed is usually not noticed by the human eye • … for example, changes in shade/detail • It produces a larger decrease in file size compared to lossless // Lossy decreases file size considerably
6(c)(i)	**1 mark** for each bullet point (**max 2**) • Value/magnitude/size of the analogue sound wave is measured a set number of times each second/time / at set intervals • Each sample/reading/measurement is given the binary number and stored in sequence
6(c)(ii)	**1 mark** for each correct point and **1 mark** for matching expansion e.g. • Decrease sample rate • … fewer samples/readings/measurements stored per second // fewer bits per second stored • Decrease sample resolution • … fewer bits per sample/reading/measurement // each sample has fewer bits • Sound outside of set/human hearing range is removed • … fewer measurements are stored / decreases the number of possible binary values so fewer bits are stored

8(a)	**1 mark** for: Denary value: 8107

8(b)	**1 mark** for each bullet point for the method (**max 2**) e.g. • Flip each bit then add 1 … • … method of converting the new binary number into denary • Most significant 1 bit is treated as the corresponding negative denary value … • … add the other positive corresponding denary values **1 mark** for correct conversion Denary value: −97
8(c)	**1 mark** each: • A logical shift moves all bits to the right and inserts zeros in the appropriate leftmost bits • An arithmetic shift moves all bits to the right but copies the sign bit into the Most Significant Bit (MSB)

7	**1 mark** each: • Working – carried values clearly indicted • Correct answer 0001 1000 • Overflow clearly indicated as overflow Example: 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1 + 0 0 0 1 1 0 0 1 (1) 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1.....(carries)

7(a)	**1 mark** for: 3300 kibibytes
7(b)	**1 mark** each: • Converting 100 to binary **0110 0100 and** 10 to binary **0000 1010** • Subtraction method - converting 10 to – 10 and adding // direct subtraction … • … correct answer **0101 1010** Method 1: Converting to -10 and adding: Binary for +10 is 0000 1010 Binary for –10 is 1111 0110 Binary for 100 is 0110 0100 100 + (-10): 0 1 1 0 0 1 0 0 +1 1 1 1 0 1 1 0 (1) 0 1 0 1 1 0 1 0 Carries: 1 1 0 0 1 0 0 0 Method 2: Direct Subtraction Borrows: 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 0 – 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0
7(c)	**1 mark** for working: 1100 0000 1111 // 2048 + 1024 + 8 + 4 + 2 + 1 // (12 * 16^2) + 15 // (12 * 16 * 16) + 15 // 3072 + 15 **1 mark** for correct answer: 3087