SMART EXAM RESOURCES
0478 IGCSE COMPUTER SCIENCE
PAPER-1
NUMBER SYSTEMS
QP-MS

1 A car park has a payment machine that allows a customer to pay for their parking.

The cost of parking is displayed as a denary number on a screen on the payment machine.

The cost of parking is stored in two 8-bit binary registers.

For the parking cost of $10.50:
- register 1 stores the denary value 10 as binary
- register 2 stores the denary value 50 as binary.

(a) Give the parking cost that would be displayed on the payment machine when the registers store:
- register 1: 00010001
- register 2: 01000110

Parking cost displayed $ ..... [2]

Working space

.....

.....

.....

.....

(b) The parking cost of $14.98 is displayed on the payment machine.

Give the 8-bit binary numbers that are stored in the registers to display the parking cost.

Register 1 .....

Register 2 .....
[2]

Working space

.....

.....

.....

.....

(c) The payment machine gives the customer a ticket when they have paid their parking cost. Each ticket has a 4-digit hexadecimal ticket number that is stored as binary.

The binary number 1010000000111101 is stored for a customer's ticket number.

Give the hexadecimal ticket number that would be displayed on this customer's ticket.

Hexadecimal ticket number ..... [4]

Working space

.....

.....

.....

.....

(d) Explain why data input into the payment machine needs to be converted to binary.

.....

.....

.....

..... [2]

(e) When a customer is leaving the car park they arrive at a barrier. The customer needs to insert their ticket into a system at the barrier. This system reads the ticket number then checks whether the parking cost has been paid for the car. The barrier is raised if it has been paid.

The system uses a microprocessor.

Describe the role of the microprocessor in the system and how it checks whether the parking cost has been paid.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

MARKING SCHEME

(a)	**One** mark for each correct part of the fee, in the correct order: – 17 – 70 (Correct fee $17.70)	**2**
(b)	**One** mark for each correct binary value: Register 1 – 00001110 Register 2 – 01100010	**2**
(c)	**One** mark for each correct hexadecimal value, in the correct order. – A – 0 – 3 – D (Ticket number A03D)	**4**
(d)	**Two** from: – It contains **logic gates/switches** … – … that process the values 1 and 0 // have two states	**2**
(e)	Any **four** from: – Compares the ticket number received to stored data – … that is a **database/file** of ticket numbers – … checks the ticket number is listed as paid – If the **data** matches/cost is paid it sends a **signal** to raise the barrier – If the **data** does not match/cost is not paid, the barrier remains down	**4**

2 The binary number 10100011 is stored in random access memory (RAM).

A logical left shift of **three** places is performed on the binary number.

(a) Give the 8-bit binary number that will be stored after the shift has taken place.

..... [1]

(b) Tick (✓) **one** box to show which statement about a logical left shift of **two** places is correct.

A It would divide the binary number by 2. ☐

B It would multiply the binary number by 2. ☐

C It would divide the binary number by 4. ☐

D It would multiply the binary number by 4. ☐

[1]

(c) 10100011 can be stored as a two's complement integer.

Convert the two's complement integer 10100011 to denary. Show all your working.

.....

.....

.....

..... [2]

(d) The binary number is measured as a byte because it has 8 bits.

State how many bytes there are in a kibibyte (KiB).

..... [1]

MARKING SCHEME

(a)	00011000	**1**
(b)	**D**	**1**
(c)	**One** mark for correct working Example: Flip and add **One** mark for correct answer: −93	**2**
(d)	1024	**1**

3 A register stores the binary number:

1	1	1	0	0	0	1	1

(a) Give the denary number for the binary number stored in the register.

..... [1]

Working space

.....

.....

.....

(b) Give the hexadecimal number for the binary number stored in the register.

..... [2]

Working space

.....

.....

.....

(c) A logical left shift of **two** places is performed on the binary number stored in the register.

Complete the binary register to show its contents after this logical left shift.

[1]

(d) The negative denary number −99 needs to be stored in the register.

Complete the register to show the binary number that would be stored, using two's complement. Show all your working.

Working space .....

.....

.....

.....

Register:

[2]

(e) The number 01001100 is added to 11100011

Add the two 8-bit binary numbers, using binary addition.

Give your answer in binary. Show all your working.

.....

.....

.....

.....

.....

..... [4]

MARKING SCHEME

(a)	– 227	**1**
(b)	**One** mark for each correct character in the correct order: – E3	**2**
(c)	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	**1**
(d)	**One** mark for suitable working method e.g. flip and add 1 **One** mark for correct answer – 10011101	**2**
(e)	**One** mark for each correct nibble (max 2) **One** mark for correct working e.g. correct carry **One** mark for showing overflow bit 1 1 1 1 0 0 0 1 1 + 0 1 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1	**4**

4 Humans use a denary number system and computers use a binary number system.

(a) Explain what is meant by a binary number system.

.....

.....

.....

..... [2]

(b) Convert the denary numbers 14, 59 and 234 to binary.

14 .....

59 .....

234 .....

[3]

Working space

.....

.....

.....

.....

(c) Convert the denary numbers 9, 26 and 65 to hexadecimal.

9 .....

26 .....

65 .....

[3]

Working space

.....

.....

.....

.....

(d) Convert the positive denary number 123 to 8-bit binary using two's complement.

Show all your working.

.....

.....

.....

..... [2]

(e) Add the binary values 00110011 and 01111000 using binary addition.

Give your answer in binary. Show all your working.

.....

.....

.....

.....

.....

.....

.....

..... [3]

MARKING SCHEME

(a)	Any **two** from: – It has a base of 2 – It only uses two **values** – … that are 1 and 0	**2**
(b)	– (0000)1110 – (00)111011 – 11101010	**3**
(c)	– 9 – 1A – 41	**3**
(d)	**One** mark for suitable working method e.g. conversion to binary **One** mark for correct answer – 01111011	**2**
(e)	**One** mark for each correct nibble (max 2) **One** mark for correct working e.g. correct carries 1 1 1 0 0 1 1 0 0 1 1 + 0 1 1 1 1 0 0 0 1 0 1 0 1 0 1 1	**3**

5 A user enters text into a computer system, using a keyboard.

An American standard code for information interchange (ASCII) character set is used to convert the text to binary.

(a) Identify **one** other character set that could be used to convert the text to binary.

..... [1]

(b) The character 'A' is represented by the denary ASCII number 65.

The character 'm' is represented by the denary ASCII number 109.

(i) Convert the **two** denary ASCII numbers to binary.

65 .....

109 .....
[2]

Working space

.....

.....

.....

.....

(ii) Convert the **two** denary ASCII numbers to hexadecimal.

65 .....

109 .....
[2]

Working space

.....

.....

.....

.....

(c) The character 'y' is represented by the binary ASCII number 01111001.

(i) Convert the binary ASCII number to denary.

..... [1]

Working space

.....

.....

.....

.....

(ii) Convert the binary ASCII number to hexadecimal.

..... [1]

Working space

.....

.....

.....

.....

(iii) A logical right shift of two places is performed on the binary ASCII number 01111001.

Give the binary number after the logical right shift of **two** places is performed.

..... [1]

Working space

.....

.....

.....

.....

(d) The character ‘T’ is represented by the binary ASCII number 01010100.

The character ‘t’ is represented by the binary ASCII number 01110100.

Add the **two** binary numbers using binary addition. Give your answer in binary. Show all your working.

.....

.....

.....

.....

.....

.....

.....

..... [3]

MARKING SCHEME

(a)	Unicode	**1**
(b)(i)	• (0)1000001 • (0)1101101	**2**
(b)(ii)	• 41 • 6D	**2**
(c)(i)	121	**1**
(c)(ii)	79	**1**
(c)(iii)	00011110	**1**
(d)	**One** mark for correct working, for example: carries **One** mark for each correct nibble. 1 1 1 1 0 1 0 1 0 1 0 0 0 1 1 1 0 1 0 0 1 1 0 0 1 0 0 0	**3**