Paper 1

30 April 2026 · 2 hours 10 minutes

Maximum: [100 marks] · Section A: Q1–10 · Section B: Q11–15

SECTION A — SHORT ANSWER

Question	Mark Scheme Answer Points	Marks
Q1 [3]	Award 1 mark each — any THREE: • Processing speed / clock speed (GHz) • RAM / primary memory capacity • Secondary storage capacity • Cost / price • Portability / physical size / weight • Power consumption / battery life	**[3]**

Question	Mark Scheme Answer Points	Marks
Q2 [2]	Award 1 mark each — any TWO: • Printer / scanner • File server / shared files and folders • Internet connection / router • Software / applications	**[2]**

Question	Mark Scheme Answer Points	Marks
Q3 [1]	Removes and returns the top/last element from the stack.	**[1]**
Q4 [1]	Checks whether the stack contains no elements; returns TRUE if empty, FALSE otherwise.	**[1]**

Question	Mark Scheme Answer Points	Marks
Q5 [3]	Award 1 mark per valid point — max 3: • Colours represented using binary / bits [1] • RGB model used: Red, Green, Blue component values [1] • Each component stored as 0–255 (8 bits); 24 bits total for true colour [1] • Greater bit depth = more colours representable [1]	**[3]**

Question	Mark Scheme Answer Points	Marks
Q6 [4]	OUT = 1 if binary number has ODD number of 1s, OR if decimal equivalent is ODD. Zero is even. Award 1 mark per 2 correct rows:	**[4]**

	`A B C | Decimal | Odd 1s? | Odd decimal? | OUT`
	`0 0 0 | 0 | No | No | 0`
	`0 0 1 | 1 | Yes | Yes | 1`
	`0 1 0 | 2 | Yes | No | 1`
	`0 1 1 | 3 | No | Yes | 1`
	`1 0 0 | 4 | Yes | No | 1`
	`1 0 1 | 5 | No | Yes | 1`
	`1 1 0 | 6 | No | No | 0`
	`1 1 1 | 7 | Yes | Yes | 1`	

Note	*OUT = 0 only for (0,0,0) and (1,1,0). Award 1 mark per 2 correct rows.*	

Question	Mark Scheme Answer Points	Marks
Q7 [4]	Starting: null ← Ali ↔ Zak ↔ Kim ↔ Mia → null Award 1 mark per correct operation:	**[4]**

Question	Mark Scheme Answer Points	Marks
	1. Delete Zak → null ← Ali ↔ Kim ↔ Mia → null	
	2. Delete first (Ali) → null ← Kim ↔ Mia → null	
	3. Delete last (Mia) → null ← Kim → null	
	4. Insert Ben at start → null ← Ben ↔ Kim → null	
Note	*Require correct bidirectional arrows and null pointers for full marks.*	

Question	Mark Scheme Answer Points	Marks
Q8(a) [1]	A set of rules/standards governing how data is transmitted/communicated between devices or systems.	**[1]**
Q8(b) [3]	Award 1 mark each — any 3: • Enable devices from different manufacturers to communicate (interoperability) [1] • Define a common format/structure for data so it is correctly interpreted [1] • Allow error detection and correction during transmission [1] • Manage data flow / prevent congestion/collision on the network [1] • Ensure security / authentication of data transmissions [1]	**[3]**
Q9 [1]	Compiler (accept: interpreter; assembler for assembly language)	**[1]**
Q10 [2]	• The process of identifying / locating errors (bugs) in a program [1] • And then correcting / removing those errors so the program works as intended [1]	**[2]**

SECTION B — EXTENDED RESPONSE

Question	Mark Scheme Answer Points	Marks
Q11(a) [2]	Award 1 mark each — any 2: • Cost of full implementation exceeds budget [1] • System does not meet original requirements / user needs [1] • Performance / speed inadequate for business operations [1] • Significant security vulnerabilities identified during evaluation [1] • Users found the interface too complex or difficult to use [1]	**[2]**
Q11(b) [4]	[+] Old system acts as backup if new system fails — reduces risk of data loss [+] Outputs from both systems compared to verify accuracy of new system [+] Staff trained on new system while business continues normally [–] Expensive — maintaining two systems simultaneously doubles costs [–] Data must be entered into both systems — duplication of effort [–] Can cause confusion for staff using two different systems at the same time	**[4]**
Note	*Evaluation must address BOTH advantages and disadvantages with development. Max 2 if only one side presented.*	
Q11(c)(i) [3]	Award 1 mark each — any 3: • Hacking / unauthorised access to the database [1] • Phishing attacks targeting employees to steal credentials [1] • Malware / ransomware infecting company systems [1] • Interception of data in transit over public networks [1] • Insider threat — malicious or careless employee leaking data [1]	**[3]**
Q11(c)(ii) [1]	Stores, sends, receives, and manages email messages on behalf of users/clients.	**[1]**

Question	Mark Scheme Answer Points	Marks
Q11(c)(iii) [2]	Award 1 mark each — any 2: • Creates a permanent written record of communication for future reference [1] • Same message sent to multiple recipients simultaneously [1] • File attachments can be easily shared [1] • Asynchronous — can be sent/received at any time without both parties present [1]	**[2]**
Q11(c)(iv) [3]	• Email travels across public networks and can be intercepted [1] • Encryption converts content into unreadable ciphertext during transmission [1] • Only the intended recipient with the correct key can decrypt and read the message [1] • Protects sensitive customer/business data from unauthorised disclosure [1]	**[3]**

Question	Mark Scheme Answer Points	Marks
Q12(a)(i) [3]	Award 1 mark per valid contrast (must address BOTH types): • Primary (RAM/ROM) directly accessible by CPU; secondary (HDD/SSD) is not [1] • RAM is volatile — loses data when powered off; secondary is non-volatile [1] • Primary is faster, smaller capacity, more expensive/GB; secondary is slower, larger, cheaper [1] • Primary stores currently running programs/data; secondary stores data permanently [1]	**[3]**
Q12(a)(ii) [4]	Award 2 marks each for TWO well-described functions: • Memory allocation: assigns blocks of RAM to programs when launched [2] • Memory deallocation: frees memory when a program terminates [2] • Virtual memory management: uses secondary storage as extension of RAM when full [2] • Memory protection: prevents processes accessing memory of other processes [2]	**[4]**
Note	*Award 1 mark if function named but not explained. Max 4.*	
Q12(b)(i) [1]	Temporarily holds data being transferred to or read from memory (RAM).	**[1]**
Q12(b)(ii) [3]	• Small, very fast memory located close to or within the CPU [1] • Stores frequently accessed or recently used instructions and data [1] • CPU checks cache before accessing slower RAM — significantly reduces access time [1] • Improves overall CPU performance by reducing memory bottleneck [1]	**[3]**
Q12(b)(iii) [4]	• Data bus: bidirectional; carries data between CPU, memory, and I/O. Width = data per cycle [1+1] • Address bus: unidirectional (CPU→memory); carries memory addresses for read/write [1+1] • Control bus: carries control signals (read/write, clock, interrupts) from CU to coordinate operations [1+1]	**[4]**
Note	*Bus names required for full marks. 1 mark per bus named + 1 mark per role described.*	

Question	Mark Scheme Answer Points	Marks
Q13(a)(i) [2]	Award 1 mark each — any 2: • GPS sensor / receiver (measures distance) [1] • Clock / internal timer (measures elapsed time) [1] • Keypad / touchscreen (driver enters extra charges) [1] • Speedometer / odometer sensor [1]	**[2]**
Q13(a)(ii) [3]	• Microprocessor receives input data from sensors (GPS, timer) [1] • Processes/calculates fare in real time by applying the preset rate formula [1] • Controls the display to output the current fare continuously to passengers [1] • Processes driver inputs for extra charges (luggage, tolls, holidays) [1]	**[3]**

Question	Mark Scheme Answer Points	Marks
Q13(b)(i) [6]	Award 1 mark per valid point — max 6: • Network of satellites orbits Earth, each transmitting position and time of transmission [1] • GPS receiver picks up signals from multiple satellites (at least 3–4 required) [1] • Receiver calculates the time taken for each satellite signal to arrive [1] • Since signals travel at the speed of light, distance from each satellite is calculated [1] • Trilateration using distances from multiple satellites determines the precise position [1] • More satellites in range = greater accuracy of position fix [1]	**[6]**
Q13(b)(ii) [2]	Award 1 mark each — any 2: • Can track taxi real-time location and estimated arrival time [1] • Fare calculated transparently/accurately based on actual distance — reduces disputes [1] • Can share journey details with others for personal safety [1]	**[2]**
Q13(b)(iii) [2]	Award 1 mark each — any 2: • Provides optimised navigation / fastest route, reducing travel time and fuel costs [1] • Accurate automated fare calculation reduces disputes with passengers [1] • Can locate nearby passengers / receive job assignments efficiently [1]	**[2]**

Question	Mark Scheme Answer Points	Marks
Q14(a) [4]	Award 2 marks each for TWO explained disadvantages (1 name + 1 in context): • Unpredictable memory usage: allocated at runtime — impossible to guarantee availability in a limited embedded system [2] • Memory management overhead: pointers/management code consume extra memory/processing time — critical when resources are severely constrained [2] • Risk of memory leaks: if memory not properly deallocated, system may run out of memory and crash [2]	**[4]**
Q14(b) [5]	Array A = [9,8,7,4,1,2,1,3] (indices 0–7). Base case: M < 3 → return 1.	**[5]**

	`rec(7,A) = A[7] x rec(6,A) = 3 x rec(6,A)`
	`rec(6,A) = A[6] x rec(5,A) = 1 x rec(5,A)`
	`rec(5,A) = A[5] x rec(4,A) = 2 x rec(4,A)`
	`rec(4,A) = A[4] x rec(3,A) = 1 x rec(3,A)`
	`rec(3,A) = A[3] x rec(2,A) = 4 x rec(2,A)`
	`rec(2,A): M=2, 2>=3 is FALSE -> return 1`
	
	`Working back:`
	`rec(3)=4x1=4 rec(4)=1x4=4 rec(5)=2x4=8`
	`rec(6)=1x8=8 rec(7)=3x8=24`
	
	`ANSWER: rec(7, A) = 24`	

Note	*Award 1 mark per correct step shown. All working required for full 5 marks.*	
Q14(c) [6]	Award 1 mark each: low/high init [1], loop condition [1], mid calc [1], comparison+found flag [1], correct low/high adjustment [1], return [1]	**[6]**

Question	Mark Scheme Answer Points	Marks

	isThere(X, ARR)
	low = 0
	high = 7
	found = FALSE
	loop while low <= high AND found = FALSE
	mid = (low + high) div 2
	if ARR[mid] = X then
	found = TRUE
	else if ARR[mid] < X then
	low = mid + 1
	else
	high = mid - 1
	end if
	end loop
	return found
	end isThere	

Question	Mark Scheme Answer Points	Marks
Q15(a) [4]	Award 2 marks each for TWO reasons (1 point + 1 explanation): • Fixed known dimensions: hall always 20 rows x 10 seats — static allocation is appropriate and efficient [2] • 2D structure mirrors physical layout: rows and columns map directly to hall rows and seats, making access intuitive e.g. CONCERT[row][seat] [2] • Char data type is appropriate: each seat holds a single character (A, B, C, or X) [2]	**[4]**
Q15(b) [4]	Award: correct loop bounds 0–9 [1], condition checking for C [1], counter increment [1], output [1]	**[4]**

	student(CONCERT, ROW)
	count = 0
	loop col from 0 to 9
	if CONCERT[ROW][col] = 'C' then
	count = count + 1
	end if
	end loop
	output count
	end student	

Q15(c) [7]	Award: outer loop rows 0-19 [1], inner loop cols 0-9 [1], check A/B/C [1 each=3], correct prices 15/10/5 [1], output total [1]. Max 7.	**[7]**

	revenue(CONCERT)
	total = 0
	loop row from 0 to 19
	loop col from 0 to 9
	if CONCERT[row][col] = 'A' then
	total = total + 15
	else if CONCERT[row][col] = 'B' then
	total = total + 10
	else if CONCERT[row][col] = 'C' then
	total = total + 5
	end if
	end loop
	end loop
	output total
	end revenue	