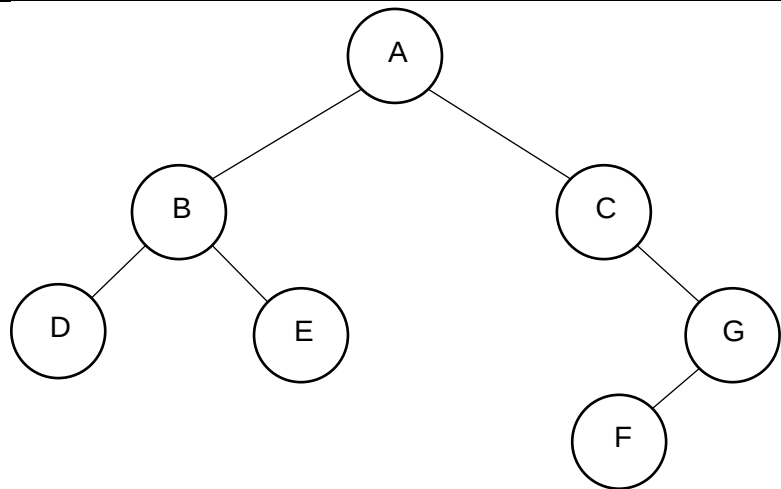


1	<table><tr><td>Condition</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Orbit around sun</td><td>T</td><td>T</td><td>T</td><td>T</td><td>F</td><td>F</td><td>F</td><td>F</td></tr><tr><td>Sufficient mass</td><td>T</td><td>T</td><td>F</td><td>F</td><td>T</td><td>T</td><td>F</td><td>F</td></tr><tr><td>Cleared neighbourhood</td><td>T</td><td>F</td><td>T</td><td>F</td><td>T</td><td>F</td><td>T</td><td>F</td></tr><tr><td>Actions</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Planet</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Dwarf planet</td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Small solar system bodies</td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td></tr></table>	Condition									Orbit around sun	T	T	T	T	F	F	F	F	Sufficient mass	T	T	F	F	T	T	F	F	Cleared neighbourhood	T	F	T	F	T	F	T	F	Actions									Planet	1								Dwarf planet		1							Small solar system bodies				1					
	Condition																																																																									
	Orbit around sun	T	T	T	T	F	F	F	F																																																																	
	Sufficient mass	T	T	F	F	T	T	F	F																																																																	
	Cleared neighbourhood	T	F	T	F	T	F	T	F																																																																	
	Actions																																																																									
	Planet	1																																																																								
	Dwarf planet		1																																																																							
	Small solar system bodies				1																																																																					
	<table><tr><td>Condition</td><td></td><td></td><td></td><td></td></tr><tr><td>Orbit around sun</td><td>T</td><td>T</td><td>T</td><td>F</td></tr><tr><td>Sufficient mass</td><td>T</td><td>T</td><td>F</td><td>-</td></tr><tr><td>Cleared neighbourhood</td><td>T</td><td>F</td><td>-</td><td>-</td></tr><tr><td>Actions</td><td></td><td></td><td></td><td></td></tr><tr><td>Planet</td><td>1</td><td></td><td></td><td></td></tr><tr><td>Dwarf planet</td><td></td><td>1</td><td></td><td></td></tr><tr><td>Small solar system bodies</td><td></td><td></td><td>1</td><td></td></tr><tr><td>Not classified</td><td></td><td></td><td></td><td>1</td></tr></table>	Condition					Orbit around sun	T	T	T	F	Sufficient mass	T	T	F	-	Cleared neighbourhood	T	F	-	-	Actions					Planet	1				Dwarf planet		1			Small solar system bodies			1		Not classified				1																												
	Condition																																																																									
	Orbit around sun	T	T	T	F																																																																					
	Sufficient mass	T	T	F	-																																																																					
	Cleared neighbourhood	T	F	-	-																																																																					
	Actions																																																																									
	Planet	1																																																																								
	Dwarf planet		1																																																																							
Small solar system bodies			1																																																																							
Not classified				1																																																																						
2(a)	Computers operate using binary code, which consists of 0s and 1s. To store, process, and transmit data, including characters, they need to be converted into a binary format that the computer can understand.																																																																									
2(b)	ASCII: Each character in ASCII is represented by a 7-bit binary value, which means it can only encode a limited set of characters and cannot represent characters from non-Latin scripts. Unicode: Unicode uses variable-length encoding, which means characters are represented using different numbers of bits, depending on the character's code point. This allows Unicode to represent a wide range of characters, including those from Asian, Middle Eastern, and other non-Latin scripts. Similarity: 1st 128 characters are the same for ASCII and Unicode Difference: Unicode uses up to 32 bits while ascii just uses 7 bits to encode the character																																																																									

2(c)	Quotient	remainder		
	125	13 (D in hex)		
	7	7		
	0			
	Answer = 7D			
3(a)	Vehicles -Reg_Code -Engine_Size -Reg_Date +set_reg_code() +set_engine_size() +set_reg_date() +get_reg_code() +get_engine_size() +get_reg_date() +calc_age() +calc_taxes() Commercial -company_type -purpose +get_company_type() +set_company_type() +get_purpose() +set_purpose() +calc_taxes() Emergency -vehicle_type -servicing date +get_vehicle_type() +get_servicing_date() +set_vehicle_type() +set_servicing_date()			

3(b)	The properties and methods in the base Vehicle class are inherited by the Commercial and Emergency classes This allows for code reusability, the inherited properties and methods do not need to be redefined in the child classes.	
3(c)	This is to ensure the attributes can only be accessed by the getter method and modified by the setter method. This would prevent accidental corruption of the attributes.	
3(d)	Define a calc_tax() in the Emergency class that always returns 0. This works by overriding the original calc_tax() method inherited from the vehicles parent class, ensuring the tax is always 0 for emergency vehicles. Or Do not define calc_tax() in the base Vehicle class. Since commercial vehicle class has its own calc_tax() method, it does not need to inherit the method from the base Vehicle class. Hence removing it will ensure emergency vehicles will not be taxed, and commercial vehicles would still defined their own calc_tax() method.	
4(a)(i)	Start: [8,5, 7, 1, 4, 2] [1, 5, 7, 8, 4, 2] [1, 2, 7, 8, 4, 5] [1, 2, 4, 8, 7, 5] [1, 2, 4, 5, 7, 8] [1, 2, 4, 5, 7, 8]	
4(a)(ii)	$O(n^2)$	
4(a)(iii)	Any list with element greater than 999 that is not already sorted. E.g. [1000, 1, 2, 3]	
4(a)(iv)	Line 5: MinVal $\leftarrow$ MyList[i] Line 6: MinValIndex $\leftarrow$ i	
4(b)(i)	Bubble sort works in place whereas merge sort does not work in place, so bubble sort requires less memory.	
4(b)(ii)	Merge sort is of time complexity $O(n \log n)$ whereas bubble sort is of time complexity $O(n^2)$. Hence when n is large, merge sort is faster.	
4(b)(iii)	A recursive sorting algorithm splits the lists into smaller lists which are sorted by calling the same algorithm. It has a base case which is the empty list or the list with one item.	
5(a)	(Bottom of stack is on the left, top is on the right) Pass 1 Stack 1: [B, C]	

	Stack 2: [A] Pass 2 Stack 1: [B, F, G] Stack 2: [A, C] Pass 3 Stack 1: [B, F] Stack 2: [A, C, G] Pass 4 Stack 1: [B] Stack 2: [A, C, G, F] Pass 5 Stack 1: [D, E] Stack 2: [A, C, G, F, B] Pass 6 Stack 1: [D] Stack 2: [A, C, G, F, B, E] Pass 7: Stack 1: [] Stack 2: [A, C, G, F, B, E, D]	
5(b)	Output: D, E, B, F, G, C, A Post-order traversal	
5(c)	It is not possible to reconstruct the tree as you do not know how many children each node has. For instance, the following tree also gives the same post order traversal	



6(a)

	Data	Pointer
[0]	" "	1
[1]	" "	None
[2]	"BOB"	5
[3]	" "	0
[4]	"SUS"	2
[5]	"PUP"	None

FreeListPointer = 3
LinkedListPointer = 4

6(b)

The data is not sorted (either in the linked list or the array).

Furthermore, it does not occupy a contiguous block in the array.

6(c)

The search pointer is initially set to the value of LinkedListPointer.

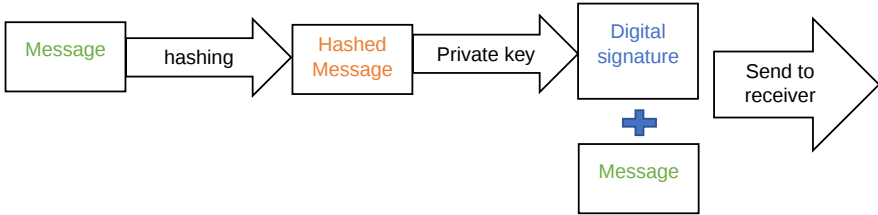
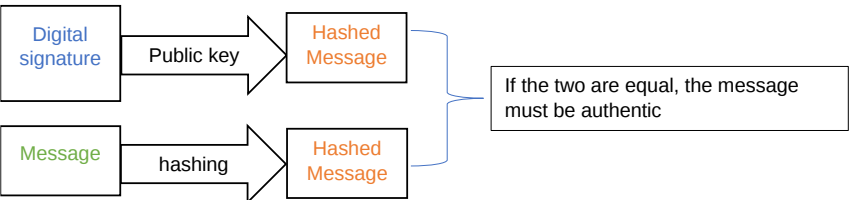
Repeat the following steps until the search pointer is None:

1. Go to the item in the array whose index is the value of the search pointer
2. If the data in that item matches the search value, return the value of the search pointer, and break the repeat loop.
3. Otherwise, set the search pointer's value to the item pointer's value.

If the repeat loop has concluded without the item being found, then it does not exist in the linked list.

7(a)	The Purchases table is not in 3NF because the composite primary key consists of CustomerID, AlbumID, and Datetime. However, the fields CustomerName and CustomerPhoneNo depend only on CustomerID.	
7(b)		
7(c)	Albums (AlbumID, AlbumName, Artiste, Genre, Price) Customers (CustomerID, CustomerName, CustomerPhoneNo) Purchases (CustomerID, AlbumID, Datetime, Quantity) 	
7(d)	<pre> SELECT AlbumName FROM Albums INNER JOIN (Purchases INNER JOIN Customers ON Purchases.CustomerID = Customers.CustomerID) ON Albums.AlbumID = Purchases.AlbumID WHERE Albums.Genre = "K-pop" AND CustomerName = "Jane Lee" ORDER BY Album.Price DESC LIMIT 1 </pre>	
7(e)	Using NoSQL means there is no fixed schema, so the database can be more flexible, e.g. allowing albums with no genre or multiple artists and genres. NoSQL is horizontally scalable so the database can be stored across multiple servers and can be readily extended when necessary, e.g. if the store expands greatly in volume.	
8(a)	Data verification it to make sure the data is accurately entered as user intended. Data validation is to check that the data entered conforms to some specification / format	
8(b)	A confirmation page where the user is shown the data he entered, and request to verify the information before clicking confirm. Double entry for important information like email address. Prompt the user to reenter his email if the 1st entry differs from the 2nd entry	
8(c)	Length check for phone number Range check for age	
8(d)	Purpose limitation: the data should only be used for the purpose of providing teaching services to the students Protection: Ensure data collected is stored securely with protection mechanisms Retention: ensure data is deleted when no longer necessary	

8(e)	Use firewalls that would filter out bad traffic. Firewalls would inspect the incoming data packets for potential malicious code, and block such traffic from entering the network. Keep software updated with latest patches. This is to prevent known malware from infecting the computer systems. Ensure data is encrypted and only authorized employees know the decryption password. So even if the data is lost, the hackers will not be able to use it.	
8(f)	1. Message is hashed and encrypted with sender's private key. 2. This forms the digital signature. 3. Original message is sent with the digital signature 4. Receiver would decrypt the digital signature with sender's public key to obtain the hashed message 5. The original message is hashed and compared with the hashed message in step 3. 6. If the two are identical, the receiver can be certain the message did come from the sender, and hence the message is authenticated.	

	Sender  Receiver 	
8(g)	Password which only the user should be aware of 2 factor authentication like code sent through SMS	
8(h)	It is a hierarchical naming system used to translate human-readable domain names (like www.example.com) into IP addresses that computers understand. DNS plays a vital role in networking by enabling users to access websites and other resources using easy-to-remember domain names instead of having to memorize complex IP addresses.	
8(i)	A MAC (Media Access Control) address is a unique identifier assigned to a network interface card (NIC) at the hardware level. It is used to identify devices within a local network. Unlike IP addresses, which are assigned by software and can change, MAC addresses are permanent and globally unique to each network device.	
8(j)	IPv4 uses 32 bits while IPv6 uses 128 bits. Hence IPv6 allows many more unique IP addresses.	