

2022 Prelim P1 sample ans

Thursday, August 18, 2022 9:04 AM

1(a)	(i)	Recursion occurs when a function calls itself as part of its execution.
	(ii)	Lines 8 and 10
	(iii)	Line 2 checks for the base case , when a is equal to b. No further recursion occurs at this point.
	(b)	↑ Return 3 hcf(15,36) ↓ ↑ Return 3 hcf(15, 21) ↓ ↑ Return 3 hcf(15, 6) ↓ ↑ Return 3 hcf(9, 6) ↓ ↑ Return 3 hcf(3, 6) ↓ ↑ Return 3 hcf(3, 3)
	(c)	When the function hcf is called, a frame is allocated for the data used by the function. The frame is pushed onto the call stack with each new function call, and control is passed to the new frame. When the function execution completes, the frame is popped off the call stack , and the return value passed to the calling frame.
	(d)	If the number of calls exceeds a certain limit, the call stack would run out of memory space . This will result in a maximum recursion depth exceeded error (or a stack overflow)

2(a)	(i)	Binary search requires a sorted array. Since the array is sorted , binary search would be more appropriate as it has a time efficiency of $O(\log n)$, compared to linear search with an efficiency of $O(n)$.
	(ii)	23, 35, 28
	(iii)	A binary search would take up to three loop iterations to search for any given element in the Figure 2 array. Hence, if the element searched for is within the first 3 indexes of the array , linear search would complete in a shorter time.
(b)	(i)	$O(n^2)$
	(ii)	The quicksort algorithm could use a random/the middle element as the pivot, which could yield a higher chance of a pivot close to the median element.
	(c)	$O(n^2)$
	(d)	Quicksort has a worst case time efficiency of $O(n^2)$ for almost-sorted arrays. Bubblesort has a best-case time efficiency of $O(n)$ (if it terminates the loop early when no swaps take place in an iteration). However, time efficiency of an algorithm only describes how the execution time scales with the amount of data, and is not an indication of speed for a particular data size . Hence, for small arrays , it is possible for bubble sort to be faster than

		quicksort.
--	--	------------

3(a)	(i)	Index	L_ptr	Data	R_ptr
		0	1	99	3
		1	2	97	-1
		2	-1	95	-1
		3	4	125	6
		4	5	121	-1
		5	-1	109	-1
		6	-1	135	-1
	(iv)	Index	L_ptr	Data	R_ptr
		0	1	109	3
		1	-1	97	-1
		2	-1	NULL	-1
		3	4	125	6
		4	-1	121	-1
		5	-1	NULL	-1
		6	-1	135	-1

3(a)	(ii)	Procedure P implements in-order traversal , which would not produce the given sequence. Pre-order traversal is required for that.																					
	(iii)	<pre> PROCEDURE P(Index : INTEGER) OUTPUT(bst_arr[Index].Data IF bst_arr[Index].L_ptr <> -1 THEN P(bst_arr[Index].L_ptr) ENDIF IF bst_arr[Index].R_ptr <> -1 THEN P(bst_arr[Index].R_ptr) ENDIF ENDPROCEDURE </pre>																					
3(b)	(i)	QueueFront: 1 QueueLast: 7																					
	(ii)	<table border="1"> <tr> <td>Index</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr> <td>Value</td><td>99</td><td>97</td><td>95</td><td>125</td><td>121</td><td>109</td><td>135</td></tr> </table>						Index	1	2	3	4	5	6	7	Value	99	97	95	125	121	109	135
Index	1	2	3	4	5	6	7																
Value	99	97	95	125	121	109	135																
	(iii)	The values in Queue are in the reverse order from its state in (i)																					
	(iv)	Queue and Stack could be implemented using a linked list which would use dynamic allocation of memory.																					
	(v)	The advantage of dynamic data structures is that they only use as much memory as needed to store the data. This results in lower memory usage, as memory need not be pre-allocated. The disadvantage of dynamic data structures is that each insertion or deletion will involve memory allocation or deallocation, which has higher overhead compared to a static data structure. (ALT: Memory use of dynamic data structures has to be monitored so that it does not consume all device memory)																					

4	(a)	Data validation checks that data being passed into a program meets a set of requirements, while data verification checks that data produced by a program
---	-----	--

		matches the expected output
	(b) (i)	Mobile number: length check Email address: format check
	(ii)	Mobile number: 999999999 is a check for extreme test value Email address: 'user@domain.name' is a check for normal test value
	(c)	<pre> FUNCTION isValid(CreditCard: STRING) RETURNS BOOLEAN Sum <-- 0 Pos <-- 16 WHILE Pos > 0 DO Digit <-- INTEGER(CreditCard[Pos]) IF Pos % 2 = 1 // Even-numbered positions: Pos is 15, 13, 11 ... THEN Digit <-- Digit * 2 ENDIF IF Digit > 10 THEN // DIV carries out integer division, truncating decimals Digit <-- Digit DIV 10 + Digit MOD 10 ENDIF Sum <-- Sum + Digit Pos <-- Pos - 1 RETURN (Sum MOD 10 = 0) ENDFUNCTION </pre>
5	(a)	<p>Authentication: Student devices can be authenticated using login mechanisms (username and password), along with a multi-factor authentication mechanism. This could involve a biometric process, or one-time password (OTP) verification.</p> <p>OR Student devices could use digital certificates to authenticate themselves on the school network. These digital certificates will have to be installed by a network administrator, and permissions set so that they cannot be tampered with.</p> <p>Authorization: The school network should implement user account controls. This will enable the school administrator to set permissions for different accounts. For example, staff accounts would be given greater access to information, while student accounts would have more limited permissions to edit information.</p>
	(b)	<p>Trojan: Trojans hide malicious code that is executed when a file is opened. These could be sent to students as an innocent-looking file attachment. When opened, the malicious code could search for sensitive information on the student learning device and send it to the attacker.</p> <p>Virus: A virus could be added to a program that the student learning device downloads and executes. The virus, once executed, could log keystrokes of the student learning device, or leak other kinds of information to the attacker.</p>
6	(a)	<p>Employee</p> <ul style="list-style-type: none"> - EmployeeID - Name - Address - DateOfBirth + Constructor() + GetEmployeeID() + GetName() + GetAddress()

		+ GetDateOfBirth() + SetEmployeeID() + SetName() + SetAddress() + SetDateOfBirth() + CalculateSalaryPayment()
		SalariedEmployee --> Employee - BaseMonthlyRate - HoursThisMonth - BonusAmount - IsOnPension + GetBaseMonthlyRate() + GetHoursThisMonth() + GetBonusAmount() + GetIsOnPension() + SetBaseMonthlyRate() + SetHoursThisMonth() + SetBonusAmount() + SetIsOnPension() ApprenticeshipEmployee --> Employee - HourlyRate - HoursThisWeek + GetHourlyRate() + GetHoursThisWeek() + SetHourlyRate() + SetHoursThisWeek()
	(b)	Child classes inherit attributes and methods from the parent class. This means they can access the attributes and methods of the parent class. For example, SalariedEmployee and ApprenticeshipEmployee will be able to access the getters and setters of the Employee class
	(c)	This describes polymorphism

7	(a)	For a table to be in 3NF, it must first satisfy 2NF, which states that all non-key attributes must be dependent on the primary key, which is (ServiceNo, Direction, StopSequence). The attributes Operator and Category do not depend on the primary key, but on the attribute ServiceNo instead.
	(b)	BusStopCode
	(c)	Bus <--1--n--> Route <--n--1--> Stop
	(d)	Bus(<u>ServiceNo</u> , Operator, Category) Route(<u>ServiceNo</u> [FK], Direction, StopSequence, BusStopCode[FK])
	(e)	Each route is unique to the bus service and direction. The route consists of a sequence of bus stops, thus each stop can only be uniquely identified as a combination of the bus service, direction, and sequence, requiring a composite key)
	(f)	SELECT Operator, COUNT(DISTINCT ServiceNo) FROM Route INNER JOIN Stop ON Route.BusStopCode = Stop.BusStopCode WHERE RoadName = 'Tampines Ave 8' GROUP BY Operator;
	(g)	Match between system and real world: The interface of the application should follow how users would typically look up bus route information. For

	<p>example, user should be able to specify a start and end location, and see all possible bus routes, instead of having to know the bus stop code.</p> <p>Visibility of system status: Users should be able to see if the application and platform are working, or down for maintenance. For example, if the database is unavailable, the main page should show that the system is down, and disable the search box.</p>
--	---