

Qn	Task
1.1	Data validation
	length check carried out
	format check carried out: string consists only of digits 0-9
	Correct return type
1.2	Checksum algorithm
	task1_1() used to validate input_value
	Correct digits doubled
	Digits correctly summed
	Correct modulus check
	Appropriately returns False and terminates program
	Appropriate returns True
1.3	sqlite3
	task1_2() used to validate credit card numbers
	correct query to update Credit_Card_VValidity: Update Users SET Credit_Card_VValidity = ? WHERE UserID = ?;
	commit() used to commit transaction
	close() used to close database connection after use
1.4	sqlite3, File IO
	correct query to fetch records from database: SELECT Name, Category_Name, SUM(Amount) AS total_transaction
	FROM Users INNER JOIN Transactions ON Users.UserID = Transactions.UserID INNER JOIN Categories ON Categories.CategoryID = Transactions.CategoryID
	GROUP BY Users.UserID, Categories.CategoryID
	ORDER BY Category_Name ASC;
	cursor used to fetch results
	data appropriately written in csv format
	close() used to close file handle after use
2.1	Library: random
	int parameter n defined, n integers generated
	correct use of randint()
	Correct return type
2.2	Sorting: mergesort
	base case correctly detected and handled
	input split into left and right subarrays
	left and right subarrays are sorted recursively
	merge: compare first elements of (sorted) left and right arrays ...
	merge: ... until either array is empty
	merge: append appropriate element to sorted list, remaining elements correctly concatenated

	Correct return value
2.3	Searching: linear search
	loop is used to iterate through input items
	Correct return value
2.4	Searching: binary search
	Start & end initialised correctly
	Middle item compared with parameter value
	Start / end appropriately updated based on comparison
	Conditional loop used to repeat above steps, with appropriate terminating condition
	Correct return value
2.5	Time complexity
	input correctly passed to functions: each function call should receive a newly-randomised list, or an unmodified copy of the original generated list
	Correct use of time module to determine execution time
	execution time determined for each approach, for 10-, 100-, 1000-, 10000-integer input
	execution times displayed in required format
	Appropriate conclusion from displayed evidence
3.1	OOP, Linked list
	Correct class and method name and syntax
	node class instantiates required attributes: - priority (should come from constructor parameter) - job (should come from constructor parameter) - nextNode (should be instantiated to None for standalone node)
	is_empty(): correctly checks for empty list, correct return value
	insert(): input parameters wrapped in a node
	insert(): node appropriately linked to previous linkedlist head
	insert(): linkedlist head is appropriately updated
	count()/displayJobs(): linkedlist correctly walked
	count(): correct return value
	displayJobs(): display job only, and total jobs at end
	Linked list correctly instantiated, items inserted in correct order using insert(), displayJobs() called
3.2	Inheritance, stack
	Stack correctly inherits LinkedList
	push(): correctly inserts data at top of stack (using insert())
	pop(): uses is_empty() or checks for missing head to check for empty list, returns None
	pop(): appropriately de-links head node, returns job
	Stack instantiated, jobs pushed in correct order
	Jobs popped from stack using pop() and displayed ...
	... using a conditional loop that uses is_empty() to check for empty stack
3.3	Sorting
	PriorityList correctly inherits Stack, overrides insert() method

	insert(): input parameters wrapped in a node
	insert(): linkedlist correctly walked ...
	insert(): ... and each node appropriately compared to new node (to ensure desc priority order)
	insert(): new node correctly linked when insertion point is determined
	PriorityList instantiated, jobs inserted in correct order
	Jobs popped from stack using pop() and displayed until empty
4.1	Programming
	game board correctly initialised
	tries correctly initialised
	mine_x and mine_y initialised (and later updated) as ints
4.2	Modularisation
	4.2 & 4.3: functions correctly defined with appropriate parameters
	randomise mine: uses randint() correctly to generate random x- and y-coordinates ...
	... and correctly updates global variables mine_x and mine_y
	prompt player: prompts player to input x- and y- coordinates using input() with an appropriate text prompt ...
	... and uses a conditional loop appropriately to validate user input and display an error message ...
	... returning the x- and y-coordinates when it passes validation
	update: x and y parameter used to check game board <i>in a consistent manner</i> i.e. (x, y) or (y, x) format
	game board correctly updated at target coordinates with an "X" or "O"
	tries correctly updated
4.3	Modularisation
	display: loop used to iterate through each row and display contents appropriately
	game over: loop used to iterate through each row and cell to check for "X" on any coordinate, returns appropriate value
	Sample program <initialise global variables> randomise_mine() while not gameover(): display(board) x, y = prompt_player() update_board(x, y) display() print("Tries:", tries)
4.4	Web programming
	Flask app correctly initialised and run() called
	HTML table used to display game board
	HTML form used to take in x- and y-coordinates
	Appropriate HTML element used to display error message
	request.form or request.args used to access player input (with appropriate processing)

	Validation carried out correctly
	Appropriate error message generated if necessary
	board is updated with validated input (using above functions)
	board and error message used to generate output (webapp response)
	defined function used to check if game is over, display number of tries in output (webapp response)
	Output: error message example shown
	Output: game board correctly displayed
	Output: game board shows example/result of gameplay
	Coding Standards
	Use indentation and white space
	Use naming conventions (e.g. meaningful identifier names)
	Write comments