Task

Recursion

Model Answer Approach

Visit our website

Copyright © 2023 HyperionDev. All rights reserved.

Auto-graded task 1

This recursive approach aims to compute the sum of elements in a list up to and including a specified index. Initially, it checks if the provided index is zero or less, in which case it returns the first element of the list. For indices greater than zero, the function recursively adds the current element to the sum of elements up to the previous index. This recursive process continues until it reaches the base case of index zero or less, at which point it returns the sum computed so far.

Each recursive call progressively builds upon the sum by including the element at the current index, thereby accumulating the total sum of elements up to the specified index. This approach effectively breaks down the problem into smaller subproblems, each focusing on computing the sum up to a smaller index. By leveraging recursion, the function provides a succinct and elegant solution without the need for explicit loops or iteration constructs.

Auto-graded task 2

This recursive solution aims to identify the largest integer within a given list. It begins by checking if the list contains only one element; if so, it returns that element. If the list is longer, it compares the first element with the largest integer found in the remaining list. This comparison is achieved through a recursive call to the same function, which continues until it reaches the base case of a single-element list. At this point, the function starts to unwind, returning the largest number found among the recursive calls.

Throughout the process, each recursive call focuses on finding the largest integer within a subset of the original list. By repeatedly subdividing the list and comparing the first element with the largest number identified in subsequent subsets, the function effectively traverses the entire list to determine the maximum value. This approach embodies the essence of recursion, utilising a divide-and-conquer strategy to break down the problem into smaller, more manageable subproblems. It avoids the need for explicit iteration constructs like loops, offering a concise and elegant solution to the task of finding the largest number in a list.

Copyright © 2023 HyperionDev. All rights reserved.