

Completing online course:

<https://runestone.academy/ns/books/published/csjava/Unit11-Recursion/toctree.html>

Submit your work through codeboard:

Cohort A: <https://codeboard.io/projects/390335>

Cohort B: <https://codeboard.io/projects/390341>

### 1.Guided

Write a recursive method called `sumOfDigits` that takes an integer as input and returns the sum of its individual digits.

The `sumOfDigits` method should perform the following steps to calculate the sum of digits:

1. Separate the given number into its individual digits.
2. Calculate the sum of the digits.
3. Return the sum as the result.

Your task is to implement the `sumOfDigits` method, using recursion, to calculate the sum of the digits of a given integer.

Example:

Given an integer `num = 12345`, the individual digits are 1, 2, 3, 4, and 5. The sum of these digits is  $1 + 2 + 3 + 4 + 5 = 15$ . Therefore, calling the method `sumOfDigits(12345)` should return 15.

### 2.Semi-Guided

You are given a two-dimensional maze represented by a character array. Each cell in the maze can contain one of the following characters:

- `.`: Indicates an empty cell that can be traversed.
- `O`: Represents the destination cell.
- `x`: Represents an obstacle that cannot be traversed.

Implement a recursive method called `explore` that takes the following parameters:

- `row (integer)`: The current row position in the maze.
- `col (integer)`: The current column position in the maze.
- `step (integer)`: The current step count.
- `maze (character array)`: The maze to explore.

The `explore` method should perform the following steps:

- Check if the current position is out of the maze boundaries. If it is, return a large value (e.g., 1000) to indicate that this path is not valid.
- Check if the current position contains an obstacle (`x`). If it does, return a large value (e.g., 1000) to indicate that this path is not valid.
- Check if the current position is the destination cell (`O`). If

- it is, return the current step count as the result.
- Mark the current position as visited by changing the value in the maze array to x.
- Initialize a variable dist with the maximum possible integer value.
- Recursively explore the neighboring cells (left, right, up, and down) by calling the explore method with updated parameters and incrementing the step count by 1.
- Restore the original value of the current position in the maze array.
- Compare the distances obtained from exploring the neighboring cells and return the minimum distance.

Your task is to implement the explore method, using recursion, to find the minimum number of steps required to reach the destination cell (0) in the maze.

### 3. Un-Guided

You are given a certain amount of money and a list of prices for different items. Your goal is to determine the maximum number of items you can buy without exceeding your available money.

Implement a recursive method called buyStuff that takes the following parameters:

1. money (integer): The amount of money you have.
2. price\_list (integer array): An array containing the prices of items.
3. idx (integer): The current index indicating the position in the price list.

The buyStuff method should return an integer representing the maximum number of items that can be bought.

Note the following constraints:

1. If the available money is less than or equal to 0, no items can be bought, so the method should return 0.
2. If the current idx is greater than or equal to the length of the price\_list, there are no more items to consider, so the method should return 0.

Your task is to implement the buyStuff method, utilizing recursion, to find the maximum number of items that can be bought given the available money.

Example:

Suppose you have money = 50 and the list of prices price\_list = [10, 20, 30, 40, 50]. By calling the buyStuff method with these parameters, you should be able to determine the maximum number of items that can be bought without exceeding the available money.