CarticaAI Testing Questions

Version 1.4
Author: Yuval W.

Table of Contents

Overview

The following document contains home-assignments for Cartica Employee candidates.

Important notes:

- Use coherent names in code.
- Make sure code compiles.
- Deliver source code only (no need for compiled versions).
- Keep OOP in mind when designing the code.
- Keep performance in mind when designing the code.
- Keep boundary cases in mind when designing the code.
- Always validate user input.
- Handle exceptions.
- Use coherent notifications and readouts.
- All applications should be repeatable (Given the same input, the application should behave exactly the same always, and produce the same output).
- Make the code as cross-platform as possible.

Instructions:

- Submissions must include working code only.
- If asked for, submissions must include test code / cases.

Weary Array Traveler

Scenario:

We want to have a function which receives an unsigned int vector (every element is a whole number >= 0) and returns a boolean answer, whether reaching the last element is possible, according to the following traversing rules:

Traversing rules:

- The first index is 0, that's where the algorithm starts.
- Algorithm may only 'jump' forward or backwards in the array according to the value in the 'current' element (e.g. if the value at index 0 is 3 the algo may only advance to index 3. if the value at index 3 is 2 – the algo may advance to both index 5 and index 1).

Directions:

1) Please write said function.

2) Please write a 'main' function which receives user input and uses this function to print an output.

3) User input may comes in 3 formats: CSV, TSV, JSON.

Examples:

[4, 4, 1, 1, 2, 2, 1000, 1] will return TRUE, since there is a route from the first element to the last element which goes: 0 (4) → 4 (2) → 2 (1) → 1 (4) → 5 (2) → 7] .

[4, 2, 1, 3, 2, 2, 1000, 1] will return FALSE, since there is no route from the first element to the last element.

Notes:

* Runtime will be a major factor in this task.

* This task can be submitted in Python / C / CPP.

* Multiple tests expected.