

**NOTE:** You may solve each problem using one of the languages within brackets. For example **[Python, C]** means that the solution can be either in Python or C.

### **[C, Python]**

Assume we have a function `get_book_info(isbn)` that takes a string [ISBN](#) argument and retrieves a struct/object containing the Title, Author, and Language of a book (each represented as a string) from a database. Write a wrapper function that increases performance by keeping some of the database results in memory for quick lookup.

To prevent memory from growing unbounded, we only want to store a maximum of N book records. At any given time, we should be storing the N books that we accessed most recently. Assume that N can be a large number when making decisions about choices of data structure(s) and algorithm(s).

### **[JavaScript/TypeScript]**

Use modern JavaScript and HTML5 to access information from the <https://restcountries.eu/> API. The goal is to display a list of all the capital cities for the country and all of its neighbouring countries. E.g. Searching for "USA" will result in a list showing "Washington, D.C.", "Ottawa", and "Mexico City".

You may assume that you have access to all ES2017 features.. *Hint:* if your solution queries a country with N neighbours, it should not then make N sequential calls to the API.

### **[HTML5-CSS-JavaScript/TypeScript]**

Implement a 5-star widget for an eCommerce site for users to record a product rating. The widget displays a horizontal row of stars that are either outlined or black, according to the product rating, from left to right. E.g. ★★☆☆☆ = rating of 3.

Multiple 5-star widgets can be present on a single page. If a user has not rated a product, the widget will have 5 outlined stars (☆☆☆☆☆). Each product on the page has a UUID.

Hovering over the Nth star will light up stars 1 to N with a grey colour (e.g. ★★★★★). Clicking a star will record the rating by sending a request to a web server with enough information to record the product and the rating. After clicking, the widget will then display the rating the user submitted with black stars (e.g. ★★★★★). Submitting the rating is handled without refreshing the page.

You may assume that you have access to all ES2017 features, and do not need to make it compatible with IE11. Assume that there is a REST endpoint setup. You may use JavaScript libraries (e.g. React, jQuery) to assist in building the widget, provided that they do not solve the entire problem. You do not need to handle AuthN/AuthZ for submitting the rating and can assume that the request includes authentication information to identify the customer (but you do need to be able to pass along the product ID).

### **[c]**

Implement a matrix-vector multiplication function in C. All elements are stored as [fixed-point numbers](#). The elements of the matrix are stored as Q7. Elements of the vector are stored as Q5.10. The result should be stored as Q5.10. Implement the function in pure C. Now assume you are running on an Arm Cortex-A with NEON SIMD (e.g. Raspberry Pi 3). Re-write the C implementation using intrinsics to optimize for speed. Measure the speedup after re-writing using intrinsics.

```
void matrix_vector_multiplication(  
    const int8_t *matrix,  
    int32_t num_rows,  
    int32_t num_columns,  
    const int16_t *input,  
    int16_t *output);
```