Solution Review: Add Main Diagonal Elements in a Matrix

Let's go over the solution review of the challenge given in the previous lesson.

We'll cover the following

- Solution
- Explanation
  - add_diagonal function

Solution

Press the **RUN** button and see the output!

```
#include <iostream>
using namespace std;

// add_diagonal function
int add_diagonal(int arr[3][3], int row, int col) {
  // Initialize sum
  int sum = 0;
  // Outer loop to traverse rows in a 2D array
  for (int i = 0; i < row; i++) {
    // Inner loop to traverse values in each row
    for (int j = 0; j < col; j++) {
      // Check if row index is equal to column index
      if (i == j) {
        // Add element at row index i and column index j in sum
        sum = sum + arr[i][j];
      }
    }
  }
  return sum;
}

// print_array function
void print_array (int arr[3][3], int row, int column){
  // Outer loop
  for (int i = 0; i < row; i++) {
    // Inner loop
    for (int j = 0; j < column; j++) {
       cout << arr[i][j] << " ";
  }
  cout << endl;
  }
}
```

```
// main function
int main() {
  // Declare variable
  int result;
  // Initialize 2D array
  int arr[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
  // Call print_array function
  print_array(arr,3,3);
  // Call add_diagonal function and store your output in result
  result = add_diagonal(arr,3,3);
  // Print the value of result
  cout << "sum = " << result ;
  return 0;
}
```

Explanation

add_diagonal function

The `add_diagonal` function takes the 2D array `arr[][]`, its `row` size, and `column` size of type `int` in its input parameters. It returns an `int` value in the output.

The main diagonal elements have the same row and column index. We iterate over each value in the matrix (2D array) and compare its row and column index. In order to iterate over a 2D array, we use nested loops. The outer `for` loop traverses rows in a 2D array. The inner `for` loop to traverse columns in each row. If the `column index j` == `row index i`, add the element in `sum`.

Let’s solve a slightly more difficult challenge related to two-dimensional arrays in the upcoming lesson.