

## Codazen Software Engineering Assessment

### Problem Statement

A Student has the following properties: first name, last name, and age. A Course can then be considered to be an array of Students that are enrolled in the Course. Given two Courses, determine how many Students are enrolled in both Courses.

### Input Format

Your function should take two Courses (arrays of Students) as inputs.

A Student object is designed as follows:

```
{  
  firstName: string,  
  lastName: string,  
  age: integer  
}
```

### Output Format

Your output should be an integer representing the number of Students that are in both Courses. Return 0 if the Courses share no Students between them. As a sanity check, your output should not exceed the smallest Course size.

### Examples

Student A: { firstName: George, lastName: Washington, age: 23 }

Student B: { firstName: John, lastName: Adams, age: 25 }

Student C: { firstName: Thomas, lastName: Jefferson, age: 27 }

Input 1	Input 2	Output
[Student A, Student B]	[Student A, Student C]	1
[Student A, Student B]	[Student A, Student B, Student C]	2
[Student A, Student B]	[Student C]	0

### Constraints

You may use any language you feel most comfortable with. However, please refrain from using any built-in/library sorting methods should you feel the need to use them. You must implement any sorting algorithm on your own.

Your implementation should consider time/space complexity.

First and last names will only contain alphabetical letters.

Ages will be nonnegative.

A Course can have any number of Students, and both Courses do not have to be the same size.

Students can have the same first name, last name or age as other Students, but no two Students can share all three of the same properties.

### Deliverables

Please include the following:

Your code containing the solution to this problem.

A set of test cases that show inputs and outputs you considered for this problem.

The time and space complexities of your solution

How many hours you worked on this assessment