Classes and Objects

A *class* defines a blueprint for an object. We use the same syntax to declare objects of a class as we use to declare variables of other basic types. For example:

```
Box box1;          // Declares variable box1 of type Box
Box box2;          // Declare variable box2 of type Box
```

Kristen is a contender for valedictorian of her high school. She wants to know how many students (if any) have scored higher than her in the 5 exams given during this semester.

Create a class named $Student$ with the following specifications:

- An instance variable named $scores$ to hold a student's 5 exam scores.
- A *void input()* function that reads 5 integers and saves them to $scores$.
- An *int calculateTotalScore()* function that returns the sum of the student's scores.

Input Format

Most of the input is handled for you by the locked code in the editor.

In the `void Student::input()` function, you must read 5 scores from stdin and save them to your $scores$ instance variable.

Constraints
$1 \leq n \leq 100$
$0 \leq examscore \leq 50$

Output Format

In the `int Student::calculateTotalScore()` function, you must return the student's total grade (the sum of the values in $scores$).

The locked code in the editor will determine how many scores are larger than Kristen's and print that number to the console.

Sample Input

The first line contains n, the number of students in Kristen's class. The n subsequent lines contain each student's 5 exam grades for this semester.

```
3
30 40 45 10 10
40 40 40 10 10
50 20 30 10 10
```

Sample Output

```
1
```

Explanation

Kristen's grades are on the first line of grades. Only 1 student scored higher than her.