Student is the derived class. Completed code for Person and a declaration for Student are provided for you in the editor. Observe that Student inherits all the properties of Person.

Complete the Student class by writing the following:

- A Student class constructor, which has 4 parameters:
  1. A string, *firstName*.
  2. A string, *lastName*.
  3. An integer, *idNumber*.
  4. An integer array (or vector) of test scores, *scores*.
- A char calculate() method that calculates a Student object's average and returns the grade character representative of their calculated average:

Grading Scale

Letter	Average (a)
O	$90 \le a \le 100$
E	$80 \le a < 90$
A	$70 \le a < 80$
P	$55 \le a < 70$
D	$40 \le a < 55$
T	$a < 40$

Input Format

The locked stub code in the editor reads the input and calls the Student class constructor with the necessary arguments. It also calls the calculate method which takes no arguments.

The first line contains *firstName*, *lastName*, and *idNumber*, separated by a space. The second line contains the number of test scores. The third line of space-separated integers describes *scores*.

Constraints

- $1 \le$ length of firstName, length of las
- length of idNumber $= 7$
- $0 \le score \le 100$

Output Format

Output is handled by the locked stub code. Your output will be correct if your Student class constructor and calculate() method are properly implemented.

Sample Input

Change Theme Java 8

```
        *   @param lastName - A string denoting the Person's last name.
        *   @param id - An integer denoting the Person's ID number.
        *   @param scores - An array of integers denoting the Person's te
        */
         public Student(String firstName, String lastName,
         int identification, int[] testScores) {
            super(firstName, lastName, identification);
            this.testScores = testScores;
        }

        /*
        *   Method Name: calculate
        *   @return A character denoting the grade.
        */
         public char calculate(){
            char score;
            int sum = Arrays.stream(testScores).sum();
            int average = sum / testScores.length;
            score = (average <= 100 && average >= 90) ? 'O' :
                    (average < 90 && average >= 80) ? 'E' :
                    (average < 80 && average >= 70) ? 'A' :
                    (average < 70 && average >= 55) ? 'P' :
                    (average < 55 && average >= 40) ? 'D' : 'T';

            return score;
        }
    }
```

Line: 51 Col: 56

Upload Code as File Test against custom input Run Code Submit Code

You have earned 30.00 points!

You are now 2 challenges away from the 3rd star for your 30 days of code badge.

75% 13/15

30 Days of Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 20:23:49

Go to Dashboard Try a Random Challenge

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5

Compiler Message

Success

Input (stdin) Download

```
Heraldo Memelli 8135627
2
100 80
```

Expected Output Download

```
Name: Memelli, Heraldo
```

Student is the derived class. Completed code for Person and a declaration for Student are provided for you in the editor. Observe that Student inherits all the properties of Person.

Complete the Student class by writing the following:

- A Student class constructor, which has 4 parameters:
  1. A string, *firstName*.
  2. A string, *lastName*.
  3. An integer, *idNumber*.
  4. An integer array (or vector) of test scores, *scores*.
- A char calculate() method that calculates a Student object's average and returns the grade character representative of their calculated average:

Grading Scale

Letter	Average (a)
O	$90 \le a \le 100$
E	$80 \le a < 90$
A	$70 \le a < 80$
P	$55 \le a < 70$
D	$40 \le a < 55$
T	$a < 40$

Input Format

The locked stub code in the editor reads the input and calls the Student class constructor with the necessary arguments. It also calls the calculate method which takes no arguments.

The first line contains *firstName*, *lastName*, and *idNumber*, separated by a space. The second line contains the number of test scores. The third line of space-separated integers describes *scores*.

Constraints

- $1 \le$ length of firstName, length of las
- length of idNumber $= 7$
- $0 \le score \le 100$

Output Format

Output is handled by the locked stub code. Your output will be correct if your Student class constructor and calculate() method are properly implemented.

Sample Input

Change Theme Java 8

```
        *   @param lastName - A string denoting the Person's last name.
        *   @param id - An integer denoting the Person's ID number.
        *   @param scores - An array of integers denoting the Person's te
        */
         public Student(String firstName, String lastName,
         int identification, int[] testScores) {
            super(firstName, lastName, identification);
            this.testScores = testScores;
        }

        /*
        *   Method Name: calculate
        *   @return A character denoting the grade.
        */
         public char calculate(){
            char score;
            int sum = Arrays.stream(testScores).sum();
            int average = sum / testScores.length;
            score = (average <= 100 && average >= 90) ? 'O' :
                    (average < 90 && average >= 80) ? 'E' :
                    (average < 80 && average >= 70) ? 'A' :
                    (average < 70 && average >= 55) ? 'P' :
                    (average < 55 && average >= 40) ? 'D' : 'T';

            return score;
        }
    }
```

Line: 51 Col: 56

Upload Code as File Test against custom input Run Code Submit Code

You have earned 30.00 points!

You are now 2 challenges away from the 3rd star for your 30 days of code badge.

75% 13/15

30 Days of Code

Congratulations

You solved this challenge. Would you like to challenge your friends?
The next challenge in this tutorial will unlock in 20:23:49

Go to Dashboard Try a Random Challenge

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5

Compiler Message

Success

Input (stdin) Download

```
Heraldo Memelli 8135627
2
100 80
```

Expected Output Download

```
Name: Memelli, Heraldo
```