Write a Java Program to solve each one of the problem. Please name your project as mentioned in the question. Submit to @homeworkbot

Dimensional Array Practice

1) Write a Java program called Average.cpp that will ask the user to enter 10 positive integers and print their sum and average in separate line

2) Write a Java program called Reverse.cpp that will store each digit of a 10-digit number in an array and print it in reverse order. //1234567897 ->7987654321

3) Write a Java program called EvenAndOddCount.cpp that will store 10 positive integer from the user and print number of even integer and odd integers

4. **Thai kickboxing**. Write a program **KickBoxer.Java** that takes an integer weight was a command line input and prints out the corresponding kickboxing weight-class according to the table below.

| weight class | from | to |
|---|---|---|
| Fly Weight | 0 | 112 |
| Super Fly Weight | 112 | 115 |
| Bantam Weight", | 115 | 118 |
| Super Bantam Weight | 118 | 122 |
| Feather Weight | 122 | 126 |
| Super Feather Weight | 126 | 130 |
| Light Weight | 130 | 135 |
| Super Light Weight | 135 | 140 |
| Welter Weight | 140 | 147 |
| Super Welter Weight | 147 | 154 |
| Middle Weight | 154 | 160 |
| Super Middle Weight | 160 | 167 |
| Light Heavy Weight | 167 | 174 |
| Super Light Heavy Weight | 174 | 183 |
| Cruiser Weight | 183 | 189 |
| Super Cruiser Weight | 189 | 198 |
| Heavy Weight | 198 | 209 |
| Super Heavy Weight | 209 | |

Use an integer array to store the weight limits and a string array to store the weight categories (ranging from Flyweight to Super Heavyweight).

Write a Java Program to solve each one of the problem. Please name your project as mentioned in the question. Submit to @homeworkbot

2D Array Problem

**Magic Squares**

Problem Description

Magic Squares are square arrays of numbers that have the interesting property that the numbers in each column, and in each row, all add up to the same total.

Given a 4 x 4 square of numbers, determine if it is magic square.

**Input Specification**

The input consists of four lines, each line having 4 space-separated integers.

**Output Specification**

Output either magic if the input is a magic square, or not magic if the input is not a magic square.

**Sample Input 1**

16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1

**Output for Sample Input 1**

magic

**Explanation for Output for Sample Input 1**

Notice that each row adds up to 34, and each column also adds up to 34.

**Sample Input 2**

5 10 1 3
10 4 2 3
1 2 8 5
3 3 5 0

**Output for Sample Input 2**

not magic

Explanation for Output for Sample Input 2

Notice that the top row adds up to 19, but the rightmost column adds up to 11.