C++ Fundamentals: Judge Assignment 2 (JA2)

The following tasks should be submitted to the SoftUni Judge system, which will be open starting **Saturday, 9 December 2017, 10:00** (in the morning) and will close on **Saturday, 23 December 2017, 23:59**. Submit your solutions here: https://judge.softuni.bg/Contests/Compete/Index/878.

After the system closes, you will be able to "Practice" on the tasks – however the "Practice" results are NOT considered in the homework evaluation.

For this assignment, the code for each task should be a single C++ file, the contents of which you copy-paste into the Judge system.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into **`int`** and any floating-point input can be stored in **`double`**.

NOTE: the tasks here are NOT ordered by difficulty level.

Task 1 – Average (JA2-Task-1-Average)

Colors in computers are often represented in the RGB format, with an 8-bit (0-255) number per each of the R (red), the G (green) and the B (blue) color channels. An example of an RGB color would be the triplet **`255, 0, 0`**, which is the color **`red`** (red channel has the highest value, the other channels are "dark" as they have 0s as values) and **`128, 128, 0`** is the **`olive`** color (yeah, I know, olives are vegetables, not colors, but that's how color experts seem to define it. It's a very dark, greenish yellow. Anyway, that's beside the point)

A common way of writing this format (e.g. in CSS) is to use the hexadecimal triplet notation, or Hex Code – we write a "#" in front of a string of 6 alphanumeric characters (letters and/or numbers), each 2 of which represent a hexadecimal number from **`0`** to **`255`**, with **`00`** being equal to **`0`** and **`FF`** being equal to **`255`**. For the above examples, **`red`** would be written as **`#ff0000`**, and **`olive`** would be written as **`#808000`**

Write a program which, given two colors written in the Hex Code format (with a "#" in front of exactly 6 hexadecimal digits), computes the "average" between the colors – by calculating the average of each channel separately – and prints the resulting color in the same Hex Code format to the console. For computing the average of the channels of two colors, just sum the numbers of the channels and divide them by 2 (integer division, i.e. round down to the nearest integer, i.e. take the floor value).

That is, if the first color has the components **`red1, green1, blue1`**,
and the second color has the components **`red2, green2, blue2`**,
then the "average" of those two colors is calculated as
`(red1 + red2) / 2, (green1 + green2) / 2, (blue1 + blue2) / 2`.
For our 2 example colors **`#ff0000`** and **`#808000`** above, the average would be **`#bf4000`**
(because **`ff = 255, 80 = 128, (255 + 128) / 2 = 383 / 2 = 191,`** which is **`bf`** in hexadecimal, for the red channel, and **`00 = 0, 80 = 128, (128 + 0) / 2 = 64`**, which is **`40`** in hexadecimal, for the green channel)

Side note: If you're interested, that's not exactly how color "averaging" is done in real systems – the correct approach would be raising each channel's value to the power of 2 before doing the average and then taking the square root of the result, but we don't want to do that in this task for the sake of simplicity, if you want to learn more, see this video: https://youtu.be/LKnqECcg6Gw

Input

Two Hex Code color values on the same line, separated by a single space. Any letters in the input will be lowercase

Output

A single Hex Code color value representing the "average" of the two colors. Any letters in the output must be lowercase.

Restrictions

The total running time of your program should be no more than **`0.1s`**

The total memory allowed for use by your program is **`16MB`**

Example I/O

Example Input	Expected Output
`#ff0000 #808000`	`#bf4000`
`#2b00b5 #0ff1ce`	`#1d78c1`