COS 126	General Computer Science	Fall 2016

Programming Exam 2

Instructions. This exam has one question. You have 50 minutes. The exam is *open course materials*, which includes the course textbook, the companion booksite, the course website, your course notes, and code you wrote for the course. Accessing other information or communicating with a non-staff member (such as via email, instant messenger, text message, Facebook, Piazza, phone, or Snapchat) is prohibited.

Submission. Submit your solution via the link on the *Class Meetings* page. Be sure to click the *Check All Submitted Files* button to verify your submission. You may submit multiple times.

Grading. Your program will be graded for correctness, clarity (including comments), design, and efficiency. You will receive partial credit for a program that correctly implements some of the required functionality. You will receive a substantial penalty if your program does not compile or if you do not follow the prescribed input/output specifications.

Discussing this exam. Discussing or communicating the contents of this exam before solutions have been posted is a violation of the Honor Code.

This exam. You must turn in this exam. Print your name, NetID, precept, and the room in which you are taking the exam in the space below. Also, write and sign the Honor Code pledge. You may fill in this information now.

Name:

NetID:

Precept:

Exam Room:

"I pledge my honor that I will not violate the Honor Code during this examination."

Signature

Problem. Write a data type `ColorHSB.java` that represents a color in *hue–saturation–brightness (HSB) format*, along with a sample client.

HSB color format. A color in HSB format is composed of three components:

- The *hue* is an integer between 0 and 359. It represents a pure color on the color wheel, with 0° for red, 120° for green, and 240° for blue.
- The *saturation* is an integer between 0 and 100. It represents the purity of the hue.
- The *brightness* is an integer between 0 and 100. It represents the percentage of white.

API specification. Your data type `ColorHSB` must implement the following API:

`public class ColorHSB`

`public ColorHSB(int h, int s, int b)`	*create a color, with hue h, saturation s, and brightness b*
`public String toString()`	*string representation of this color*
`public boolean isGrayscale()`	*is this color a shade of gray?*
`public int distanceSquaredTo(ColorHSB that)`	*squared distance between the two colors*
`public static void main(String[] args)`	*a sample client (see below)*

Here is some more information about the expected behavior of each method API:

- *String representation:* return a string composed of the integers for hue, saturation, and brightness (in that order), separated by commas, and enclosed in parentheses. Here is an example:

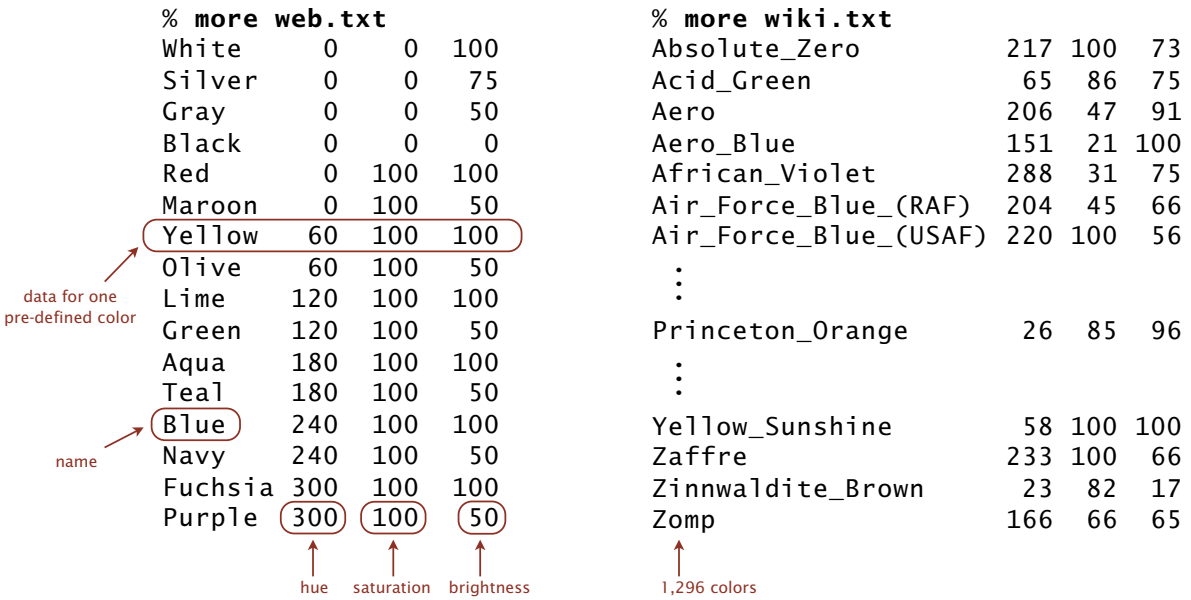
  `(26, 85, 96)`

- *Grayscale:* a color in HSB format is a shade of gray if either its saturation or brightness component is 0% (or both).
- *Distance:* the squared distance between two colors (h_1, s_1, b_1) and (h_2, s_2, b_2) is

  $$\min\left\{(h_1 - h_2)^2\ ,\ \ (360 - |h_1 - h_2|)^2\right\} \quad + \quad (s_1 - s_2)^2 \quad + \quad (b_1 - b_2)^2$$

  For example, the squared distance between $(350, 100, 45)$ and $(0, 100, 50)$ is $10^2+0^2+5^2 = 125$.
- *Exceptional situations.* For simplicity, assume that each constructor argument is in its prescribed range and that the argument to `distanceSquaredTo()` is not `null`.
- *Sample client:* your program should take three integer command-line arguments h, s, and b; read a list of pre-defined colors from standard input; and print to standard output the pre-defined color that is nearest to (h, s, b).

Input specification. The input from standard input consists of a sequence of one or more lines. Each line contains a string (the name of a pre-defined color) and three integers (its hue, saturation, and brightness components), separated by whitespace.



The data files `web.txt` and `wiki.txt` are available via the *Class Meetings* page.

Output specification. The output to standard output consists of one line: the name of the nearest pre-defined color and the string representation of that color, separated by whitespace.

```
% java-introcs ColorHSB 25 84 97 < web.txt
Red (0, 100, 100)

% java-introcs ColorHSB 350 100 45 < web.txt
Maroon (0, 100, 50)

% java-introcs ColorHSB 25 84 97 < wiki.txt
Princeton_Orange (26, 85, 96)
```

Do *not* print any other output to standard output.

Restrictions. *You should not need to declare arrays, stacks, queues, or symbol tables.*

Submission. Submit `ColorHSB.java` via the link on the *Class Meetings* page.