## **Requirements:**

You have a bag filled with jelly beans of three different colors: pink, green, and yellow and 3 cups **that cannot be moved** arranged in a line in front of you. The end result should be that all the jelly beans have been removed from the bag and placed into cups. Each cup should only contain 1 color. The jelly beans should be placed into the cups so that the pile that is the smallest is on the left, and the pile that is the largest is on the right. Jelly beans can not be placed anywhere except in the bag or in a cup.

**Notes:**

* For the sake of this, I decided, to define the colors with a hexidecimal value, the reference for these values is on this url: (“<https://www.w3schools.com/cssref/css_colors.asp>”)
* I used relatively simple coding parameters, I also used fictional predetermined classes, e.g. getColor(), which is preforming a pull method to check what color we are currently working with. With working with only 3 colors, going in order, it will first see if we are working with pink, e.g. #FFC0CB, if not then it moves on to the next color, so on and so forth. This is my fictional loop and of course if the getColor() returns that no color is present then it will exit this loop.

**Step 1: Define Data (Variables):**

pink = **#**FFC0CB;

green = **#**008000;

yellow = **#**FFFF00;

cupOne;

cupTwo;

cupThree;

myBag = pink + green + yellow;

myCount = currentCount == newCount;

newCount;

countDown = myCount.getColor() – 1;

newValue;

**Step 2: Instructions (Functions): //In this particular case I decided to use an if then statement to get all the current values of each color of jelly bean, then I decided to use two, if...else if...else statements, one to see what the largest value is, and one to see what the smallest value is. Then I compared the two values, by taking the smaller one and subtracting it from the larger one, giving me the middle cup (cupTwo). Then returning the values in order from left to right.**

**Solution:**

//Finds the values of all the current colors:

**if (newCount.pink > 0) {**

myBag.pink.countDown == 0;

get.value(pinkValue);

**}** **then (newCount.green > 0) {**

myBag.green.countDown == 0;

get.value(greenValue);

**} then (newCount.yellow > 0) {**

myBag.yellow.countDown == 0;

get.value(yellowValue);

**} end;**

//Checks to see which value is the largest number

**if (pinkValue > greenValue && pinkValue > yellowValue) {**

cupOne = pinkValue;

**} elseif (greenValue > pinkValue && greenValue > yellowValue) {**

cupOne = greenValue;

**} else (yellowValue > pinkValue && yellowValue > greenValue){**

cupOne = yellowValue;

**}**

//Checks to see which value is the smallest number

**if (pinkValue < greenValue && pinkValue < yellowValue) {**

cupThree = pinkValue;

**} elseif (greenValue < pinkValue && greenValue < yellowValue) {**

cupThree = greenValue;

**} else (yellowValue < pinkValue && yellowValue < greenValue){**

cupThree = yellowValue;

**}**

**cupTwo = cupOne – cupThree;**

//The placement of the return is left, middle, right:

**return (cupOne, cupTwo, cupThree);**