Homework 1: Coloring!  
Due 23 May, Friday

For this assignment you will be modifying the Sausage Man image shown on the first day of class. You will create a **ColoringPicture** class that extends the **Picture** class (located in your java-source folder). In the class you will have a method called **colorIn()** that changes all the yellow in the Sausage Man image to another color. The Sausage Man picture is provided for you. The change should look like the images below.

![Before and After images](image.png)

Your class should include:
- A constructor that calls the super class’s constructor
- A method **colorIn()** that changes the yellow pixels to another color (in the above case, red)
- A main method that creates an instance of **ColoringPicture** with the Sausage Man image, modifies it using the **colorIn()** method, and then displays the new image.

Hints:
- You should use **FileChooser.getMediaPath()** to access the image
- Use the incorporated **Color** class to compare the different colors (**Color.YELLOW**). To use this class, include the statement **import java.awt.*;** at the beginning of your class
- When comparing colors, you can either compare each red, blue, and green value separately, or just use the **equals()** method

WARNING:
- If your code does not compile, **you will receive a zero for the assignment**
- If your code includes a call to **FileChooser.setMediaPath**, **you will receive a zero for the assignment**
- **Make sure to turn in the .java file**, NOT the .java~ or the .class files
- **You will lose points if your code is not well commented**
- Do not forget to include your name and an appropriate collaboration statement. Failure to do so will result in a loss of points.

Files to turn in
- ColoringPicture.java

Extra Credit
- For extra credit, write a second method called `fillHead()` that fills in only Sausage Man’s head, including his nose. For full credit, use recursion (you may end up with two methods as a result, one to find where the head starts, and another to actually change the pixels).
- Another option for extra credit (worth less than `fillHead()`), is to make a `colorIn()` method that takes two `Color` inputs, the `oldColor` to change from, and the `newColor` to change to. Calls to this should look like:
  `pic.colorIn(Color.RED, Color.WHITE)` where Sausage Man is originally red and we want to make him white.
  o NOTE: Because of the way the colors in the image are set up, `Color.BLACK` will not work properly (for some reason, it’s not quite pure black).
  o If you do this extra credit, you should still have the original required form of `colorIn()` as well.
  o If for some reason the inputted `oldColor` is not already in the picture, then the method should perform no action (the picture should look the same as it did before the method call).