The goal of this third Python programming assignment is to write more of your own code inside a provided program framework, but this time involving graphics. We will be exploring several functions provided by JES (requestIntegerInRange, makeEmptyPicture, show, repaint, and addLine), as well as some color constants used by JES (red, green, blue, yellow, etc.). You may wish to look at the documentation for these functions in the Guzdial book and using the JES help system before attempting this assignment.

In the JES environment, type in the following program code exactly as you see it here and save the program as Lab3.py in your Python folder. Where you see my name in the comment code, replace it with your own name. As always, be very careful about indentation and capitalization. Where you see the symbols open up several blank lines; this is where you will be writing your own code at a later time.

```
# William T. Verts - Lab #3

def MakeImage(NewWidth, NewHeight, NewSeparation):
    MyCanvas = makeEmptyPicture(NewWidth, NewHeight, yellow)
    show(MyCanvas)
    return

def Run():
    W = requestIntegerInRange("Enter Width [1..1024]", 1, 1024)
    H = requestIntegerInRange("Enter Height [1..768]", 1, 768)
    N = requestIntegerInRange("Enter Steps [1..50]", 1, 50)
    MakeImage(W, H, N)
    return

In JES click the Load Program button. At the >>> prompt, type Run() with the parentheses and press Enter. The program should run. Fix any syntax errors or other mistakes.

When the program runs, it will ask for three numbers representing the width and height of an image, and a third number to be explained later. Enter 320 for the first number, 240 for the second number, and anything for the third. If all works correctly, you will see a 320×240 pixel, yellow box appear on screen. Do not proceed until this works.
Next, in the blank space you opened up in the \texttt{MakeImage} function, write new code to paint a set of horizontal blue lines followed by a set of vertical red lines, with the separation between lines determined by the \texttt{NewSeparation} parameter. For example, if you entered 320 for the width, 240 for the height, and 25 for the steps, your program should create the following image:

To paint lines, you will need to use the JES \texttt{addLine} function, which has the following format:

\begin{verbatim}
addLine(picture, startX, startY, endX, endY, color)
\end{verbatim}

Your code must paint the blue horizontal lines first, starting at \texttt{Y=0} (the top row of pixels on the screen), and then the red vertical lines next, starting at \texttt{X=0} (the left column of pixels on the screen). The red lines must cross over top of the blue lines.

\textbf{Your code must be general enough} to allow a user to enter any legal values for image width, image height, and number of steps, and have the program create the correct image. Think about loops! Finally, at the end of your \texttt{MakeImage} function, before the \texttt{return}, insert a call to the JES \texttt{repaint} function to update the image on the screen. The format is:

\begin{verbatim}
repaint(picture)
\end{verbatim}

Test your program with several different values for the width, height, and separation. When you are finished and everything runs correctly, submit your program using the on-line form at: \url{http://people.cs.umass.edu/~verts/cmpsci119/Lab_Submitter.html} (please make certain to select Lab 3 from the drop-down list).