Problem 2: Edge Detection

Once we have a gray-scale image we can (try to) perform edge detection on it. The result is a black-and-white image, where all edges are white and everything else black. I.e., there are only two colours. Edge detection sounds hard, but it is relatively easy. An edge occurs when there is a large and sudden contrast between two regions of an image. E.g., the black print on this white page. Hence, we want to find all the pixels that are right between the regions of high contrast, and these pixels (may) represent edges in the image.

We will use the following definition of a pixel "on an edge": If $p(x, y)$ is a pixel at location (x, y), then this pixel is on an edge if it differs (significantly) from any of the pixels below it or right of it. That is, we can use the following formula

$$d(x, y) = \sqrt{(p(x, y) - p(x + 1, y))^2 + (p(x, y) - p(x, y + 1))^2}$$

where, if $d(x, y)$ is greater than some threshold t, then $p(x, y)$ is part of an edge, otherwise, it is not. That is, in the resulting black-and-white image, a pixel $q(x, y) = 255$ (white) if $d(x, y) > t$, else $q(x, y) = 0$ (black). The threshold value, t, is typically set by the user. A higher value for t requires a higher contrast between two regions to be considered an edge, meaning some edges will be missed. A lower value of t requires a lower contrast between two regions, but will detect false edges.

Write a program called `Problem2.java` that reads in gray-scale image and generates a black-and-white image representing the edges in the original image. The program should load the image using the provided `ImageRW` code (see Hints and Suggestions in the file `a2.pdf`), perform edge detection, generate a black-and-white image where white pixels denote edges, and then write the resulting image to the specified output file (using the provided `ImageRW`) code.

Input

The names of the input and out image files, separated by a white-space, followed by an integer representing the threshold to be used for edge detection.

Output

The program produces the specified image file and does not print anything to the console. You can view the output file using a standard image viewer on your computer.

Sample Input	Sample Output
`ball_gs.png ball_ed.png 50`	*N/A*