A. In our image processing assignment, we used four kernels to detect particular kinds of edges. For example, one kernel detected *horizontal* edges, where the color value increased across the edge, as we moved from *top to bottom*. Now, design a 3 × 3 kernel that detects *diagonal edges, that run from top-right to bottom-left*, where the color value increases across the edge as we move from *top-left to bottom-right*.

B. Here is our `mappedImage` code, without comments. List the *types* of the expressions given.

```
def mappedImage(image, f):
    width = image.getWidth()
    height = image.getHeight()
    newImage = cImage.EmptyImage(width, height)
    for row in range(height):
        for col in range(width):
            pixel = image.getPixel(col, row)
            rgb = [pixel.getRed(), pixel.getGreen(), pixel.getBlue()]
            newRGB = f(rgb)
            newPixel = cImage.Pixel(newRGB[0], newRGB[1], newRGB[2])
            newImage.setPixel(col, row, newPixel)
    return newImage
```

a. `f`:

b. `width`:

c. `row`:

d. `pixel`:

e. `rgb`:

f. `f(rgb)`:

C. What is the running time, in $\mathcal{O}$-notation, for a list of length n? Show detailed work.

```
# Sorts a list of numbers, by altering the list in place. Returns the list.
# Input: List of numbers. Output: List of numbers.
def insertionSortInPlace(l):
    for i in range(1, len(l)):
        temp = l[i]
        hole = i
        while hole > 0 and l[hole - 1] > temp:
            l[hole] = l[hole - 1]
            hole -= 1
        l[hole] = temp
    return l
```

D. In $\mathcal{O}$-notation, what is the efficiency of the *encryption* step of the RSA cryptosystem? Define any variables that you introduce, and explain your answer.

E. Write the following function recursively. (Yes, it's from the homework.)

```
# Returns the sum of two lists of numbers (of equal length >= 0).
# Input: List of numbers. List of numbers of the same length.
# Output: List of numbers of the same length.
def listSum(a, b):
```

F. Write the following function recursively.

```
# Returns the binary numeral for the given number, as a list of bits.
# Input: Nonnegative integer. Output: List of 0s and 1s.
def bits(n):
```

G. In a typical graphical user interface for computers, the user can initiate actions by clicking on-screen buttons with the mouse. For example, when a button labeled "Print" is clicked, this button calls upon the program's `printDocument` function, to begin printing. Think about what attributes an object of class `Button` might have, and write the class' initialization method.