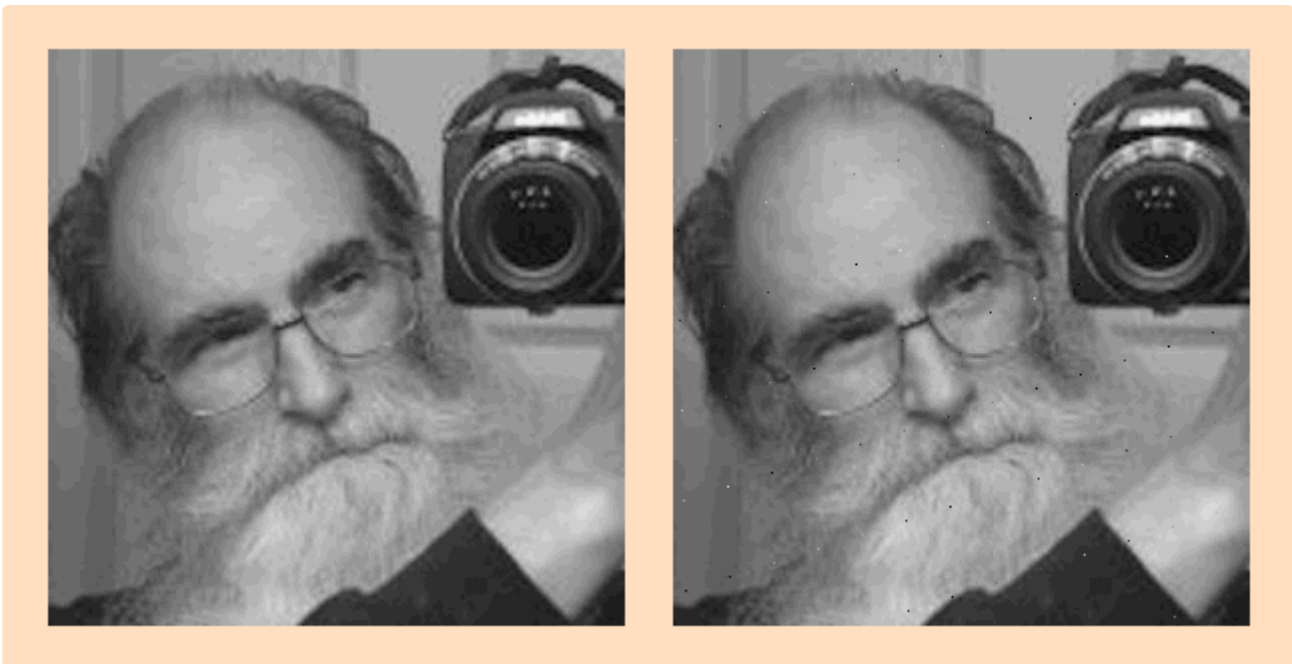


Hiding in Plain Sight

One of these pictures contains a hidden message 83 characters long:



Externally and internally, however, they are both very similar. Externally, they are what you see above. Internally, each of them is just a very long string of characters, where each character corresponds to a pixel in the image. There are just 27 allowed characters (numbers 64 to 90 on the ASCII chart):

@ABCDEFGHIJKLMNOPQRSTUVWXYZ

“@” means black, and “Z” means white, with the letters in-between representing increasingly lighter shades of gray. The definition of the image on the left looks like this in our program:

The cipher is a string equal to “NNNNNOOOOOOOPP...”

It’s much longer, of course. 82,944 bytes, to be exact. This is the routine we use to draw it on the screen:

To display a cipher in a box (as a picture):

Put the cipher’s first into a byte pointer.

Put the box’s left-top into a spot.

Loop.

If the spot’s y is the box’s bottom, break.

Put the byte pointer’s target into a byte.

Convert the byte to a color.

Draw the spot with the color.
Add 1 to the byte pointer. If the byte pointer is greater than the cipher's last, break.
Move the spot right 1 pixel.
If the spot's x is the box's right,
put the box's left into the spot's x; move the spot down 1 pixel; repeat.
Repeat.
Refresh the screen.

And this is the routine that we use to encode a message into a cipher image:

To encode a message in a cipher given a key:
Put the message's first into a message byte pointer.
Put the cipher's first into a cipher byte pointer.
Loop.
If the message byte pointer is greater than the message's last, break.
Add the key to the cipher byte pointer.
If the cipher byte pointer is greater than the cipher's last, break.
If the message byte pointer's target is the space byte,
put the at-sign byte into the cipher byte pointer's target.
If the message byte pointer's target is not the space byte,
put the message byte pointer's target into the cipher byte pointer's target.
Add 1 to the message byte pointer.
Repeat.

We thus replace some of the bytes in the cipher image's string with the characters of the hidden message. Since the message affects only a very small percentage of the image's characters, and since the characters in the message come from the same subset of the ASCII chart, and since we choose a "key" that will spread the message as thinly as possible across the entire image, the visual effect is very hard to detect. In this case, there are 83 characters in the message. Can you spot 83 differences in the images above?

To get the message back out, we use this routine:

To decode a cipher into a message given a key:
Clear the message.
Put the cipher's first into a byte pointer.
Loop.
Add the key to the byte pointer.
If the byte pointer is greater than the cipher's last, break.
If the byte pointer's target is the at-sign byte,
append the space byte to the message; repeat.
Append the byte pointer's target to the message.
Repeat.

So let's find out what that hidden message says. Here's the test routine:

To run:
Start up.
Clear the screen to the tan color.
Imagine a box 3 inches by 3 inches in the middle of the screen.
Move the box left 1-5/8 inches.
Display the cipher in the box (as a picture).
Encode the message in the cipher given 997.

Move the box right $3\frac{1}{4}$ inches.
Display the cipher in the box (as a picture).
Decode the cipher into a decoded message given 997.
Write the decoded message below the box.
Refresh the screen.
Wait for the escape key.
Shut down.

Note that, in Plain English, “ $1\frac{5}{8}$ inches” means “one and five-eighths inches”, not “one *minus* five eighths inches”. Likewise for “ $3\frac{1}{4}$ ”.

Note also that we use 997 for the key because the message is 83 characters long, which is roughly $\frac{1}{1000}$ of the 82,944 characters in the image, and 997 is the largest prime less than 1000.

The output on the screen looks like this:



GREETINGS EARTHLINGS THIS TEXT IS EVIDENCE OF AN INTELLIGENT EXTRATERRESTRIAL AGENT

If we want the “defects” in the encoded cipher to be less noticeable, we can use a shorter message or a larger picture.

On a personal side note, I must say I’m struck by the quality of the images. After all, we’re using a trivial data structure (a string) and only 27 shades of gray. Who wudda thunk?