Programming Assignment #4 — Due: March 14 (Saturday) at 11:59 PM

Steganography is the process of concealing a secret message, image, or video within another message, image, or video. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention.

Start the assignment by downloading and extracting `hw4.zip` from:

http://www.cs.uni.edu/~fienup/cs1120s15/homework/

For this assignment I will be giving you two pictures: `barbara.jpg` and `embeddedMsgBarbara.png`. Your job is to extract the secret message by comparing the two pictures. The secret message was embedded by the supplied `embedMsg.py` program. It embedded the secret message into the image as follows:

1) Assign each character to a numeric character code:
- the letters 'A' to 'Z' have character codes 1 to 26,
- the letters 'a' to 'z' also use codes 1 to 26 (so letter-case information in the message is lost),
- the ' ' (space) character has character code 0,
- the '.' (period) character has character code 27, and
- the '?' (question mark) character has character code 28.
- all other characters have character code 29.

2) Starting at pixel x=0, y=0 and proceeding row-wise across the picture `barbara.jpg`, the RGB values are adjusted to embed the sequence of characters in the secret message, i.e., first character embedded at x=0, y = 0, second character at x=1, y = 0, etc.

3) To make as small of a color change to a pixel, the character code's value is split "evenly" across each R, G, and B. For example, to embed the letter 'J' with character code of 10, we:
   a) use integer division of 10 / 3 to get 3,
   b) adjust R by 3 and G by 3,
   c) adjust B by the rest, i.e., (10 - 3 - 3) or 4

NOTE: to avoid going outside the range of values 0 - 255 when adjusting the RGB values, values below 128 will be adjusted upward and values above or equal to 128 will be adjusted downward. For example, `barbara.jpg` at pixel x = 0, y = 0 has an RGB value of (168, 131, 105) and the first letter of the secret message is 'T' (character code of 20), so the red adjustment is -6, the green adjustment is -6, and the blue adjustment is 8. Thus, the resulting x = 0, y = 0 pixel of `embeddedMsgBarbara.png` has an RGB value of (162, 125, 113).

Remember the above part is already done for you in the `embedMsg.py` program. Your job will be to finish the partial program `decryptMsg.py` which decrypts the secret message by comparing the two pictures: `barbara.jpg` and `embeddedMsgBarbara.png` at corresponding pixels. You'll need to complete the function:

```
def decryptMsg(picture, pictWithMsg):
  """ Decrypts the secret message by comparing the two pictures and
      returns the charList containing the secret message. """
  charList = []  # an empty list initially
  #your for-loops to iterate over the pictures and decrypting the secret message
.
.
     # append each decoded character to the end of charList by:
     charList.append(decodedChar)
.
.
  # last line of decryptMsg that returns the charList to the main function
  return charList
```

Hints:

- You'll probably want to use the absolute value function, `abs`, on each difference of the RGB values of the two pixels.
- When converting from a numeric character code back to a character, you can take a brute force approach using a big `if-elif-else` statement. HOWEVER, you can handle all the character codes for the letters (1 to 26) by adding `(ord('A') - 1)` to get the ASCII value for the letter, and then using the `chr` function which returns the corresponding character given an ASCII value.
- The decoded secret message is written to the file `decodedMsg.txt` by the `decryptMsg.py` program, so you can look at it to determine if your program is working.

The steps for the homework submission system are:

1. Design, write, debug, and test your programs in a folder called `hw4`. "Zip" the folder `hw4` into a single file called `hw4.zip` (In Windows, right click on the `hw4` folder and select `Send to | Compressed (zipped) folder`)
2. Log on to the submission system at: **https://www.cs.uni.edu/~schafer/submit/which_course.cgi**

   (It is very likely that you will get some security certificate warnings when trying to use this. You may add an exception and accept the existing security certificate.) Use the same CatID user-name and password you use to log on the lab computers.
3. Select the course and section number of "CS 1120, Media Computation, Fienup". Click the "Continue".
4. Select the homework that you wish to submit: "HW 4: Steganography program". Click the "Continue" button.
5. Specify how many extra files you want to submit. Just leave it at 0. Click the "Continue" button.
6. Upload your programs by Browsing and selecting your `hw4.zip` file. Click the "Continue" button.
7. The next page reports on the status of the upload(s). You can always continue to upload a better version of the program until the deadline. The newer file will replace an older file of the same name.

(If you miss the deadline, you'll need to submit it as above, but select "Late Homeworks" in step 4 above.)