COMP 8505 ASSIGNMENT TWO TESTING DOC

Yiaoping Shu + Anthony Figueroa

A00930347 October 4 2018

Usage

Before running the program, ensure you are the root user and have the zip folder stored somewhere on your computer. Navigate to the zip/tar file and extract it to a location of your choice. Go to the extracted files location.

Ensure each file can be ran by performing a chmod on the files for reading the file.
Ensure that the pictures that you want to use are in the “pictures” folder. The new images will appear in the “new” folder.

To run the program, type in the following command with the following switches.

#python3 util.py -m [mode] -c [cover image] -s [secret image] -n [new file name]

In the above case, use encode for the mode to encode the image.

Ensure the extension is typed for each file image and name.

Mode: Can choose between ‘encode’ or ‘decode’, which will either encode the image or decode the image

Cover image: The name of the file that you want to have your secret data hidden inside.

Secret image: The name of the file or image that you want to hide your data.

New file name: The name of the file to write to for the new cover image that contains the secret data

After running and encoding the image, the image should show up in the pictures/new folder. After inspecting the image, you can decode the image with the following arguments:

#python3 util.py -m decode -c [cover image] -n [new file name]

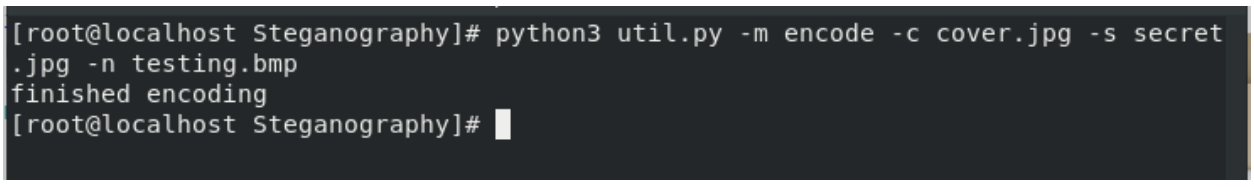
Testing

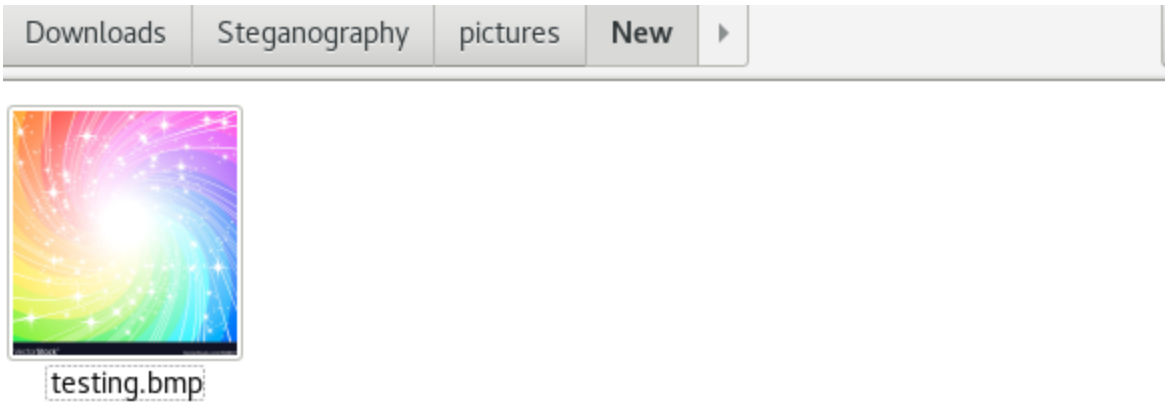
Test #	Description	Expected Result	Result (Pass/Fail)
1	**Navigate to source folder. Execute program by typing in util.py**	**User can execute util.py**	**Pass**
2	**Navigate to source folder. Execute program in encode mode with cover and secret image incorrect**	**User should be notified that image is not there**	**Pass**

3	**Execute program in encode mode with bmp image as cover and secret, with out image as bmp for next 2 steps as well**	**Out image is displayed as bmp format**	**Pass**
4	**Execute program in encode mode with png image as cover and secret**	**Out image is displayed as bmp format**	**Pass**
5	**Execute program in encode mode with jpg image as cover and secret**	**Out image is displayed with bmp extension**	**Pass**
6	**Execute encode steps 3, 4, and 5, with the out image as bmp**	**Out img is displayed as bmp format**	**Pass**
7	**Execute encode steps 3, 4, and 5, with the out image as png**	**Out image is displayed in png format**	**Pass**
8	**Decode the image by putting new secret image as png from step 7**	**User can see decoded image in the new folder**	**Pass**
9	**Decode the image by putting the new secret image as bmp from step 6**	**User can see decoded image in the new folder**	**Pass**

Screenshots

Test #1



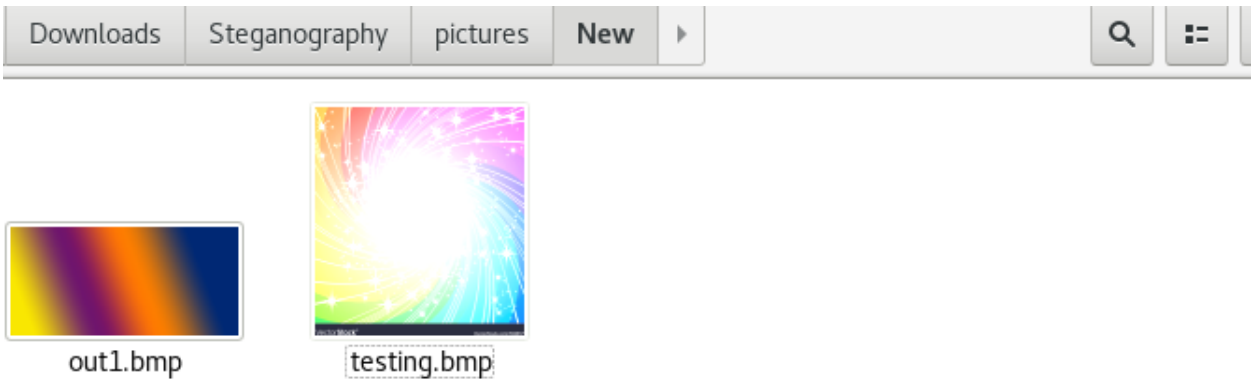


Test #2

```
FileNotFoundError: [Errno 2] No such file or directory: 'pictures/coverNone.jpg'
```

Test #4

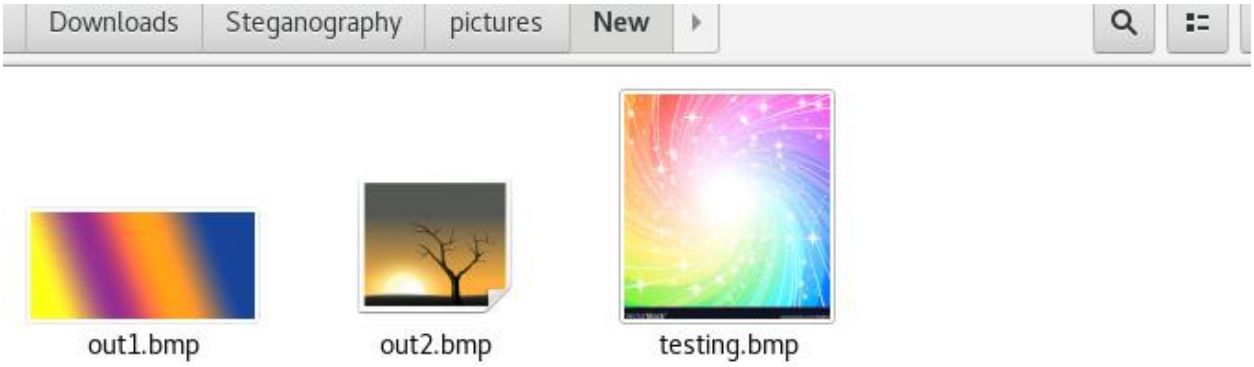
```
[root@localhost Steganography]# python3 util.py -m encode -c cover1.png -s secre
t1.png -n out1.bmp
finished encoding
[root@localhost Steganography]#
```



Test #5

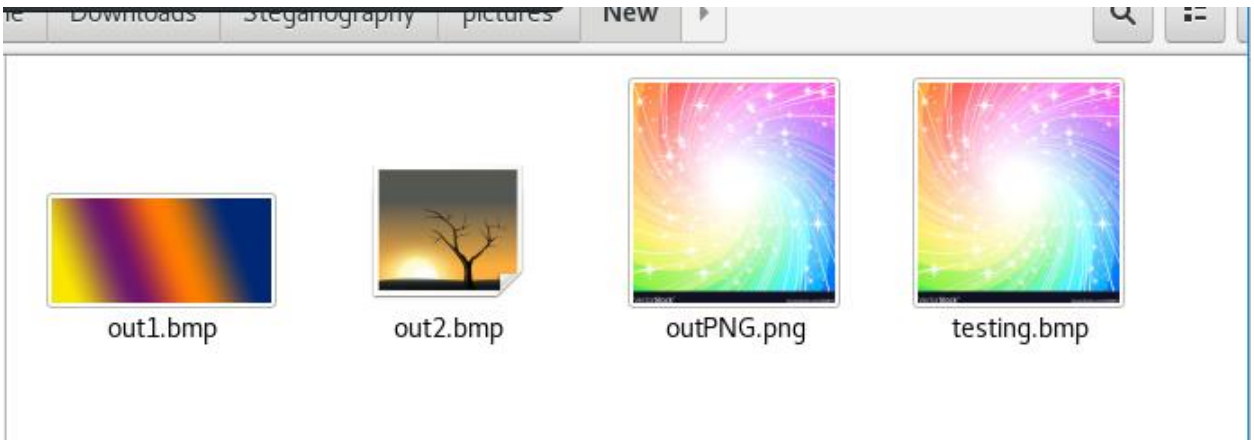
```
^C[root@localhost Steganography]# python3 util.py -m encode -c cover.jpg -s secr
.jpg -n out2.bmp
finished encoding
[root@localhost Steganography]#
```

Test #6

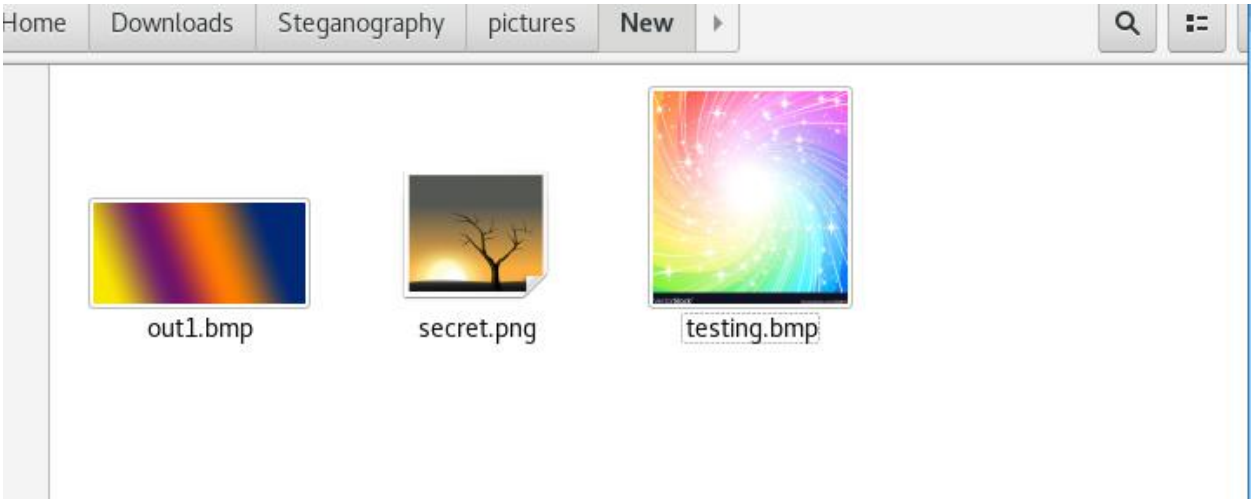


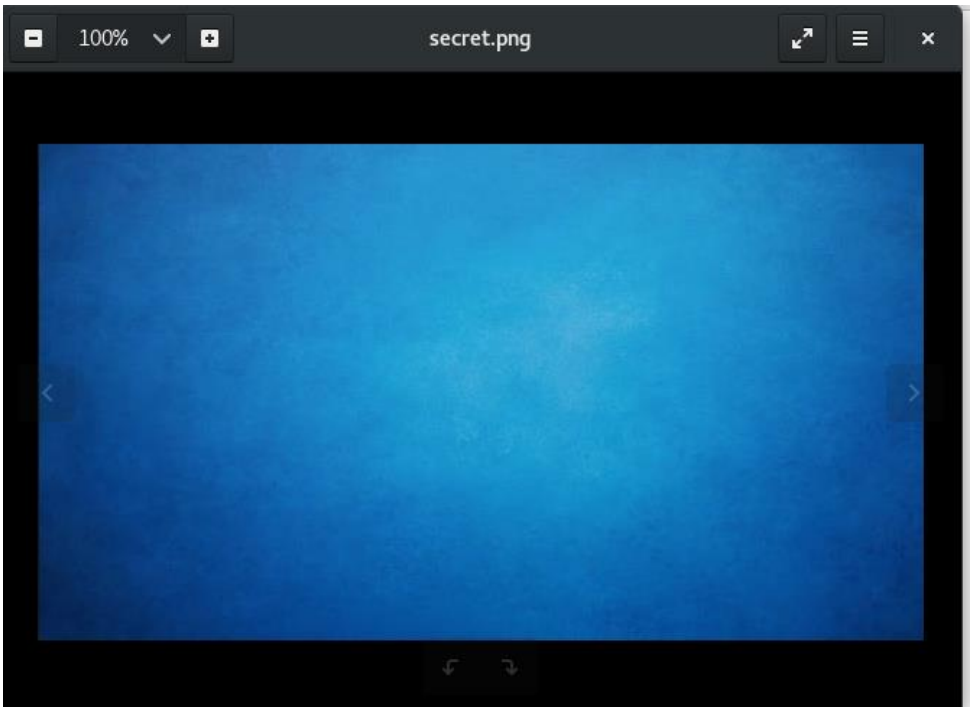
Test #7

```
[root@localhost Steganography]# python3 util.py -m encode -c cover.jpg -s secret
.jpg -n outPNG.png
finished encoding
[root@localhost Steganography]#
```



Test #8


Home
Downloads
Steganography
pictures
New
out1.bmp
secret.png
testing.bmp


100%
secret.png

Test #9

```
[root@localhost Steganography]# python3 util.py -m decode -c outPNG.png -n UnSec
ret.jpg
Starting to decode
Writing to file...
Finished decoding
[root@localhost Steganography]#
```