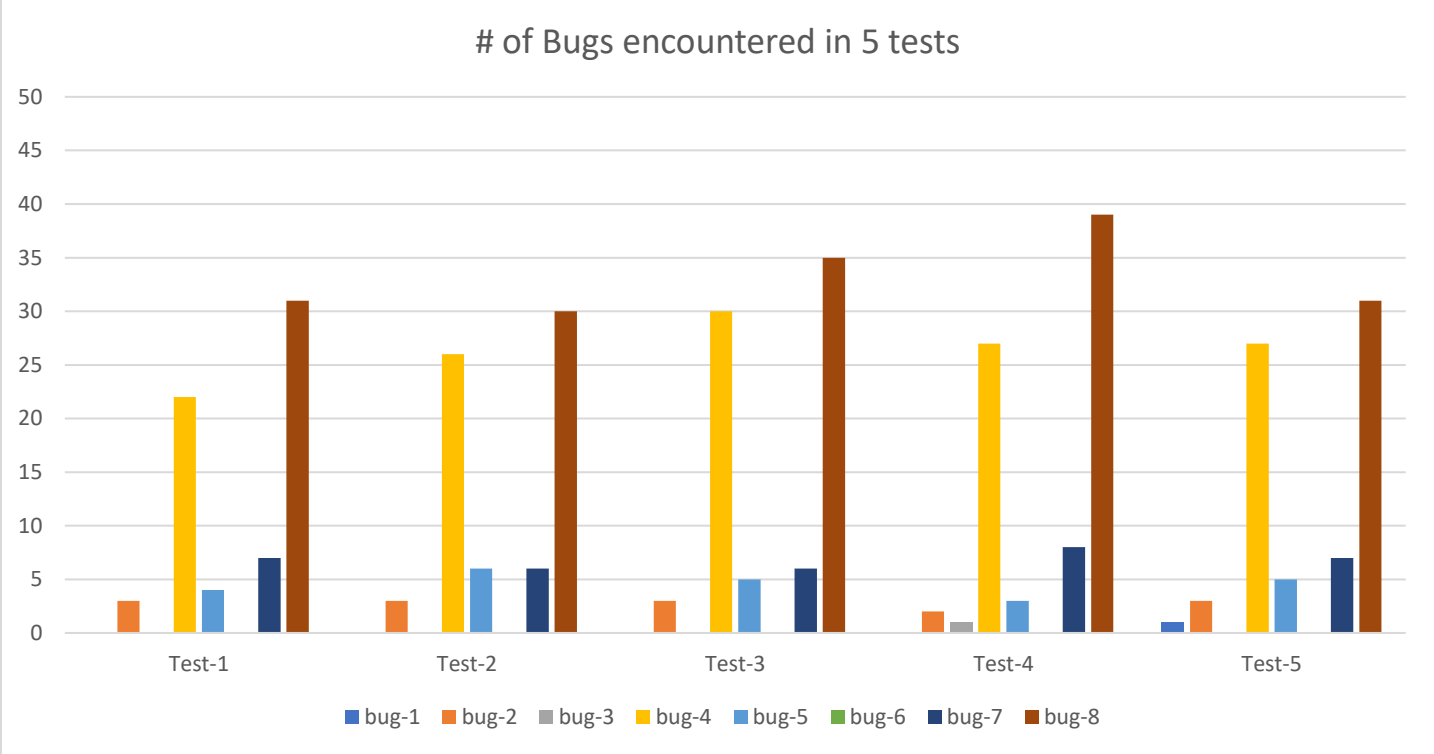
Programming Assignment-2: Fuzz-Testing

1. Fuzzer design overview:
    - Fuzzer is created using python language with consideration that python 3.6.9 version is installed on EUSTIS machine.
    - It runs the test 808 times which is the length of byte array of the provided jpg file to catch 8 bugs introduced in jpg2bmp executable by inputting different jpg files into jpg2bmp executable.
    - It also deletes any file that did not cause bug in order to save storage space.
    - Finally, Simple message indicating bug number, file name and how many times bugs occurred is printed.
    - To run the provided fuzzer.py python file, please follow the steps mentioned in section 7.
2. Process of creating modified jpg:
    - Fuzzer first imports jpg file and converts it into byte array. The image file is imported only once in the whole test to reduce number of import commands.
    - After creating a byte array, it is assigned to a new byte array which is used to create modified image.
    - The process of creating modified jpg files is performed in a loop 808 times to create different files. Hence, the original jpg file is kept intact otherwise this process would fail.
    - With each loop iteration, one of element's value modified with random number.
    - The reason for modifying only one element in every jpg file, is because 7 out of 8 bugs were found with this method. On the other hand, bug #5 was not encountered while modifying multiple elements in the same jpg file.
    - Some modified image files returned error code without encountering any of the 8 bugs.
    - In the last step, the modified byte array is converted back into "jpg" file and exported.
    - Naming convention for modified jpg file is "testing-x.jpg", where "x" is the loop iteration number.
3. Executing and detecting bug
    - Once modified jpg is created, jpg2bmp is called with modified jpg as input.
    - The Fuzzer waits for executable to exit, if the bug occurred message is printed on standard out. Message also indicated the Jpg file name that caused the bug.
4. Saving storage space
    - Because of the limited storage in EUSTIS machine, fuzzer also deletes any jpg file that did not cause any bug.
    - It does this by checking exit code of jpg2bmp
    - This process occurs with each loop iteration, hence larger number of tests are possible to run on smaller storage space.
5. Results from running the fuzzer 5 times with every run performing 808 tests.

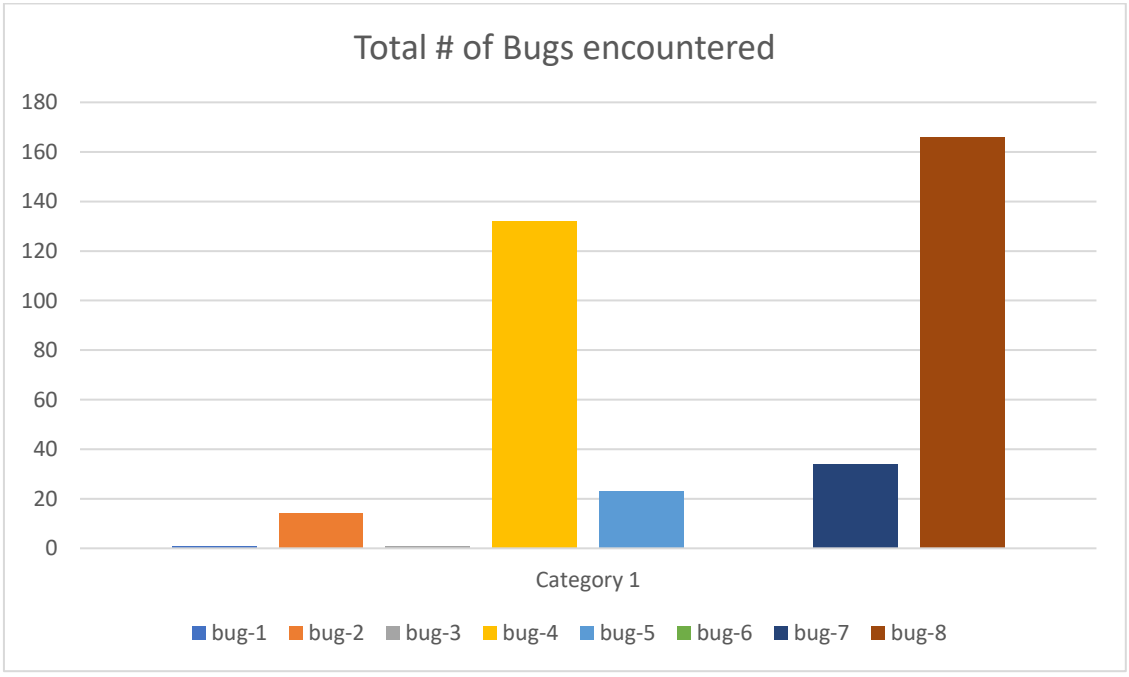
	Test-1	Test-2	Test-3	Test-4	Test-5	Total
Bug-1	0	0	0	0	1	1
Bug-2	3	3	3	2	3	14
Bug-3	0	0	0	1	0	1
Bug-4	22	26	30	27	27	132
Bug-5	4	6	5	3	5	23
Bug-6	0	0	0	0	0	0
Bug-7	7	6	6	8	7	34
Bug-8	31	30	35	39	31	166

6. Graph comparing the bugs encountered in 5 fuzzing tests:



- From the table and graph, it's clear that bug #4 and bug#8 have maximum frequency with the corresponding fuzz test design. While bug #1 is encountered only once in the 5th test and bug #6 is not encountered at all.

- Graph bellow shows comparison of total number of times each bug is encountered across 5 tests.



7. To run the fuzzer.py python file, please type the following command in terminal without the double quotes, “Python3 fuzzer.py”
    - Please make sure that jpg2bmp, cross.jpg and fuzzer.py are in the same folder.
    - Example screenshot of successfully running fuzzer.py:

```
ra996142@net1547:~/project2$ python3 fuzzer.py
Bug #1 triggered.
FuzzInputNum = 48, fileName: testing-48.jpg
Bug #7 triggered.
FuzzInputNum = 187, fileName: testing-187.jpg
Bug #7 triggered.
FuzzInputNum = 188, fileName: testing-188.jpg
Bug #7 triggered.
FuzzInputNum = 189, fileName: testing-189.jpg
```

8. Screenshots of each bug encounters with the corresponding modified jpg file as input:
    - When executing jpg2bmp to test the modified jpg files, please make sure that jpg2bmp and modified image files are in the same folder.

```
ra996142@net1547:~/project2/bugged-jpeg$ ./jpg2bmp testing-bug-1.jpg test.bmp
Bug #1 triggered.
Segmentation fault (core dumped)
ra996142@net1547:~/project2/bugged-jpeg$ ./jpg2bmp testing-bug-2.jpg test.bmp
Bug #2 triggered.
Segmentation fault (core dumped)
ra996142@net1547:~/project2/bugged-jpeg$ ./jpg2bmp testing-bug-3.jpg test.bmp
Bug #3 triggered.
Segmentation fault (core dumped)
ra996142@net1547:~/project2/bugged-jpeg$ ./jpg2bmp testing-bug-4.jpg test.bmp
Bug #4 triggered.
Segmentation fault (core dumped)
ra996142@net1547:~/project2/bugged-jpeg$ ./jpg2bmp testing-bug-5.jpg test.bmp
Bug #5 triggered.
Segmentation fault (core dumped)
ra996142@net1547:~/project2/bugged-jpeg$ ./jpg2bmp testing-bug-7.jpg test.bmp
Bug #7 triggered.
Segmentation fault (core dumped)
ra996142@net1547:~/project2/bugged-jpeg$ ./jpg2bmp testing-bug-8.jpg test.bmp
Bug #8 triggered.
Segmentation fault (core dumped)
ra996142@net1547:~/project2/bugged-jpeg$
```