Once you have identified two distinct bugs in the csvParser application, answer the following

questions using 100 words or less per question or subquestion:

1. Which fuzzers did you select for running a BFF campaign? Why did you choose these

fuzzers? For full credit, make sure to base your answer of “why” on the behavior of each

fuzzer. “I selected every fuzzer because I wanted to test them all” is not the type of

answer we are expecting.

2. Excluding the verify fuzzer, what are some fuzzers that are ineffective at finding a crash?

Why do you think these fuzzers were ineffective at finding crashing test cases?

3. Using the data generated by the BFF campaigns and other debug utilities you ran as a

guide, examine the source code in csvParser.cpp and answer the following questions

for the first bug that you found in the csvParser application:

a. Which line of code in csvParser.cpp contains the bug?

b. Explain why the failing input discovered by the BFF caused the application to

crash.

c. How could you fix the bug so that the program will execute properly?

4. Using the data generated by the BFF campaigns you ran as a guide, examine the source

code in csvParser.cpp and answer the following questions for the second bug that you

found in the csvParser application:

a. Which line of code in csvParser.cpp contains the bug?

b. Explain why the failing input discovered by the BFF caused the application to

crash.

c. How could you fix the bug so that the program will execute properly?