1. Defect report: Write a short report that describes what the valgrind trace means.

From the Valgrind trace we are able to see what type of error the code is experiencing. The output on the regular code hints at there being unitialized values. Specifically, we are pointed towards “file_info” of type *unz_file_info64* and then the char array “filename_inzip”.

As it is now, an uninitizaled variable can lead to having random content within which makes that accessing it would give faulty data. For example, “file_info” is expecting data corresponding to the factors such as size and date. In order to fix this we can initizalize this to a set value, {0} for example that would ensure there not being unknown data when reading it.

```
        char filename_inzip[256];
        unz_file_info64 file_info;
```

This can be patched by simply doing:

```
        char filename_inzip[256] = {0};
        unz_file_info64 file_info = {0};
```

With this, Valgrind reports :
ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

3.2 Fuzzing

1. We created a simple txt-file with some random letters, we then zipped it.
2. It took 4 times.
3. I think it is relatively easy and effective to find bugs with fuzzing as the fuzzing approach is random and is not prone to human subjectivity where before you test a program can miss many errors due to only focusing on specific areas of the program.