

EECS678  
Hao Luo  
Sep 13<sup>th</sup>

## Lab 2 report

After I finished others parts which were included in the lab session, I firstly checked the prgram with valgrind command “--leak-check=full”. Then I found there was a invalid free function call in line 150, so I deleted the line “free(bug\_info.useless\_bug);”. Besides of this one, there was also a bug “invalid read of size 1” in line 145. When I read this line, I found the string “bug\_info.sentence[2]” has been “freed” from storage before the printf function call. Therefore, I just changed this one with “NULL”. Now, there was only one remaining memory leak. I checked the line 101, and it was about the initialization of “bug\_info.sentence[2]”. In my idea, it should be related with problem that reallocating memory without deleting the original one. By reading the related lines about “bug\_info.sentence[2]”, I found that “bug\_info.sentence[2]” has been changed in line 127 without calling the free function. I called the free function before this line, and then all the memory leaks has been removed. However, I was still not able to to pass the “valgrind test”. I read the output file, and I saw my output was not satisfied with the formly output “The most colorful bug is a a is bug colorful most The ”. So I checked with the “echoohce” function. There was already a for loop to print the strings backwardly, but now it seemed not useable. In this loop, it had set the “NULL” pointer as the front of the array, and it was the reason why it didn’t work. I fixed this problem by swaping it with “stop\_begining”, also removing the dereference before the “\*iter” in order to keep them in the same type. Now, all the bugs has been solved in this lab.

This lab so was informative that I learned GDB and Valgrind were quitely useful tools in debugging. Actually, the final part was such a challenge for me because although this bug existed, the program was still running perfectly without memory leaks. Comparing with GDB, I feel easier to use Valgrind. But I’ll try to study and practice the way of using GDB in my future debugging.