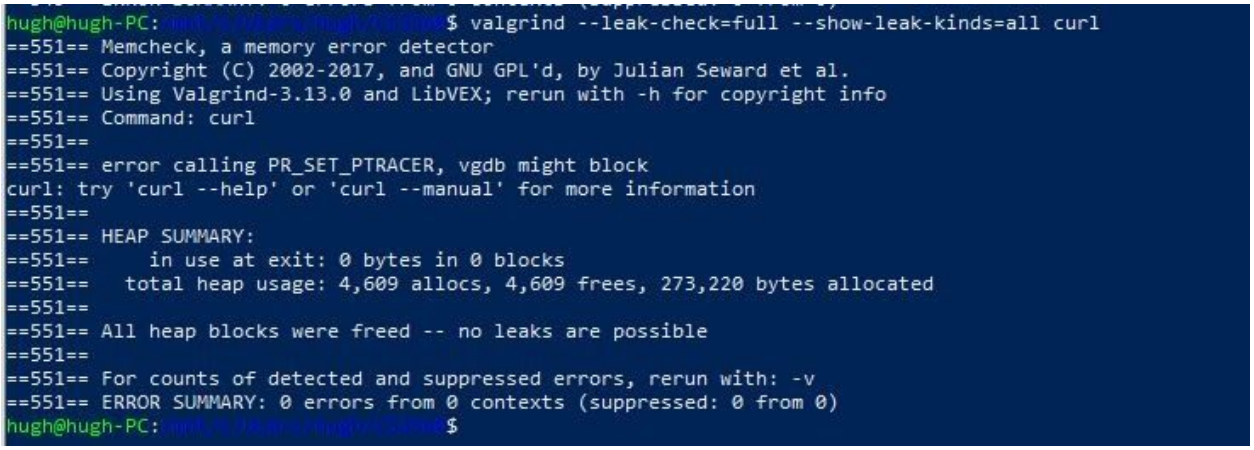
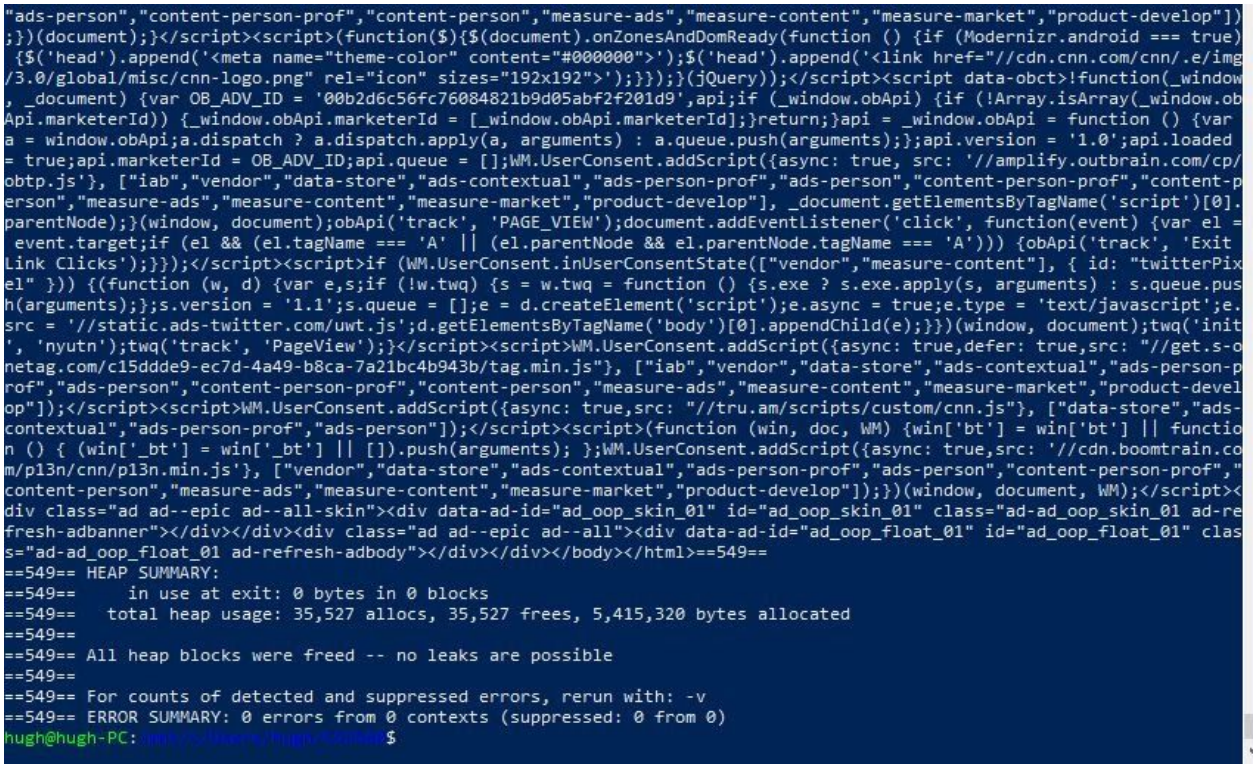
1. I checked curl on WSL by using: valgrind --leak-check=full --show-leak-kinds-all curl

This did not find any memory leaks of any sort with the curl command, and the same thing happened when checking curl https://www.cnn.com, but above the summary section of valgrind a lot of output was shown in powershell.





Using Valgrind on the word count command, it seems the program has no issues at all with memory and output the correct numbers.

```
hugh@hugh-PC: $ valgrind --leak-check=full --show-leak-kinds=all wc index.html
]==554== Memcheck, a memory error detector
==554== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==554== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==554== Command: wc index.html
==554==
==554== error calling PR_SET_PTRACER, vgdb might block
    183   30879 1131411 index.html
==554==
==554== HEAP SUMMARY:
==554==     in use at exit: 0 bytes in 0 blocks
==554==   total heap usage: 233 allocs, 233 frees, 18,827 bytes allocated
==554==
==554== All heap blocks were freed -- no leaks are possible
==554==
==554== For counts of detected and suppressed errors, rerun with: -v
==554== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
hugh@hugh-PC: $ ]
```

6)

Using cppcheck on program.cc, it seems that no matter what I tried cppcheck could't find any errors, but I received the error that cppcheck could not find the include files.

```
hugh@hugh-PC: $ cppcheck --enable=all program.cc
Checking program.cc ...
(information) Cppcheck cannot find all the include files (use --check-config for details)
```

Using valgrind on the program, there were no errors or memory leaks found.

```
hugh@hugh-PC: $ valgrind ./count
==1484== Memcheck, a memory error detector
==1484== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==1484== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==1484== Command: ./count
==1484==
==1484== error calling PR_SET_PTRACER, vgdb might block
1 Lines.
6 Characters.
==1484==
==1484== HEAP SUMMARY:
==1484==     in use at exit: 0 bytes in 0 blocks
==1484==   total heap usage: 2 allocs, 2 frees, 73,216 bytes allocated
==1484==
==1484== All heap blocks were freed -- no leaks are possible
==1484==
==1484== For counts of detected and suppressed errors, rerun with: -v
==1484== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

7 and 8)
Unfortunately I could not figure out what was wrong with my test file or include paths as I was getting the errors resulting below, I did not have enough time to figure them out, sorry.