CIS 330: Project #4B  
Assigned: June 4th, 2017  
Due: June 9th, 2017  
Note: no late penalty will be applied to this project, as long as it is submitted by  
June 13th (meaning 6am on June 14th). No work will be accepted after this time,  
even for half credit.

Worth 4% of your grade  

Please read this entire prompt!

Assignment:  
Make your 3E project run memory error and leak free

Tasks:  
1) start with your 3E source code. (If you use 3F or beyond, the exceptions and  
logger make it hard. That said, if you have already done this and don’t want  
to backtrack to 3E, then you can use a later version of your code)  
2) run valgrind on your 3E program (you may need to use ix to do this ... esp.  
since Mac+valgrind doesn’t always work well)  
3) look at memory errors and memory leaks  
4) fix errors and leaks ... repeat until valgrind declares your program leak free  
and memory error free

Submit:  
1) a screenshot of the valgrind output (see mine below)  
2) your source code

```
hank@ix:~/3F/3F 65$ valgrind proj3F ~/3A_input.pnm 3F_output.pnm  
==16125== Memcheck, a memory error detector  
==16125== Copyright (C) 2002-2011, and GNU GPL'd, by Julian Seward et al.  
==16125== Using Valgrind-3.7.0 and LibVEX; rerun with --h for copyright info  
==16125== Command: proj3F /home/users/hank/3A_input.pnm 3F_output.pnm  
==16125==  
==16125== HEAP SUMMARY:  
==16125==     in use at exit: 0 bytes in 0 blocks  
==16125==     total heap usage: 33 allocs, 33 frees, 108,022,422 bytes allocated  
==16125== All heap blocks were freed -- no leaks are possible  
==16125== For counts of detected and suppressed errors, rerun with: --v  
==16125== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 2 from 2)
```  

Notes:
1) if I had memory errors, they would have occurred after “Command: proj3F” and before “HEAP SUMMARY”. None there, so I’m OK.
   a. Don’t forget to use your 3E code … not 3F code. (unless you know a later version is OK)
2) Re memory leaks: “All heap blocks were freed” is the magic statement … that means no memory leaks
3) Due to a bug in the C++ library, there is likely going to be a leak. I always get a leak. Others are reporting no leak. When I run their code, I still see a leak. So if you do see a leak, it is OK. That said, it should be exactly one leak, for exactly 72,704 bytes. Screenshot below.

hank@ix-xenial: ~/private/SB/Proj3E 26$ valgrind proj3E 3A_input.pnm out.pnm
==13091== Memcheck, a memory error detector
==13091== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==13091== Using Valgrind-3.11.0 and LibVEX; rerun with --h for copyright info
==13091== Command: proj3E 3A_input.pnm out.pnm
==13091==
==13091== in use at exit: 72,704 bytes in 1 blocks
==13091== total heap usage: 43 allocs, 42 frees, 117,532,149 bytes allocated
==13091==
==13091== LEAK SUMMARY:
==13091==   definitely lost: 0 bytes in 0 blocks
==13091==   indirectly lost: 0 bytes in 0 blocks
==13091==   possibly lost: 0 bytes in 0 blocks
==13091==   still reachable: 72,704 bytes in 1 blocks
==13091==   suppressed: 0 bytes in 0 blocks
==13091== Rerun with --leak-check=full to see details of leaked memory
==13091==
==13091== For counts of detected and suppressed errors, rerun with: --v
==13091== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)