Homework:

(You may spend ~20 hours for this homework)

1. (90 pts) Write down dynamic NPD (Null-Pointer-Dereference) check tool using Clang based on the provided template C++ file.
   A. When NPD occurs, a C program terminates its execution with only "Segmentation fault (core dumped)" message w/o useful information for debugging. The goal of this HW is, if your <*-npd.c> file causes NPD, to print out the crashing line and crashing expression information.
   A. Your NPD check tool should receive a single preprocessed file (*.i) and generates instrumented version <*-npd.c> file.
      i. A Preprocessed C file can be obtained by gcc -E <filename>.c -o <filename>.i
   B. Instrumented target programs should detect NPD in the 3 NPD example C files (in the examples directory). Note that your NPD check tool should instrument a target C file in a general way (i.e., instrumentation method should not be overfitted to 1-npd.c, 2- npd.c, 3-npd.c).
      ii. (30 pts) Instrumented 1-npd.c should print "`NPD at line 1.c:28, *nullptr`" and terminate.
      iii. (30 pts) Instrumented 2-npd.c should print "`NPD at line 2.c:13, intptr4[0]`" and terminate.
      iv. (30 pts) Instrumented 3-npd.c should print "`NPD at line 3.c:22, aptr->f2[0]`" and terminate.
      v. *-npd.c files should print the crash information into stderr.
2. (100 pts) Apply your NPD check tool on the attached buggy version of grep.i file and submit your result. The crashing test input is "./grep-npd -n 'if' grep.c".
3. (10 pts) Measure runtime overhead of your instrumented program (all examples and grep). (i.e. Compare the execution time of the original program and the instrumented program.)

Note 1. TA will evaluate your code with provided example files and fault-inserted grep.i files.
Note 2. Please refer attached slides to get more hint on the implementation.
Note 3. You do not have to check NPD on function pointers.