1. (90 pts) Write down NPD (Null-Pointer-Dereference) check tool using Clang based on the provided template C++ file.
    A. Your NPD check tool should receive a single processed 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. Normal C program normally just prints "Segmentation fault (core dumped)" message and terminates execution when NPD happens. Your objective is to print out the crashing line and expression when your <*-npd.c> file resulted in NPD.
    C. 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).
        i. (30 pts) Instrumented 1-npd.c should print "NPD at line 1.c:34, *nullptr" and terminate.
        ii. (30 pts) Instrumented 2-npd.c should print "NPD at line 2.c:17, intptr4[0]" and terminate.
        iii. (30 pts) Instrumented 3-npd.c should print "NPD at line 3.c:25, aptr->f2[0]" and terminate.
        iv. *-npd.c files should print the crash information into stderr.
2. (100 pts) Apply your NPD check tool on attached buggy grep.i file and submit your result. The crashing test input is "./grep-npd -n 'if' grep.c".
3. (10 pts) Measure overhead of your instrumented program (all examples and grep). (i.e. Compare the execution time of the original program and your 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 don't have to check NPD of function pointers.