

Homework:

(You may spend ~20 hours for this homework)

1. (90 pts) Write down `kcov-branch-identify` using Clang based on the provided template C++ file. `kcov-branch-identify` receives a file name of a single C file and prints the list of the branches at source code level as they are and the total number of branches of the C file.

See the following output for the attached `example-kcov.c`:

```
$ ./kcov-branch-identify example-kcov.c
function: f2
  If ID: 0   Line: 4   Col: 2   Filename: ./example-kcov.h
function: f1
  If ID: 1   Line: 19  Col: 2   Filename: example-kcov.c
function: main
  If ID: 2   Line: 30   Col: 2   Filename: example-kcov.c
  If ID: 3   Line: 32   Col: 9   Filename: example-kcov.c
  For ID: 4   Line: 40   Col: 2   Filename: example-kcov.c
  While ID: 5 Line: 45   Col: 2   Filename: example-kcov.c
  Do ID: 6   Line: 50   Col: 2   Filename: example-kcov.c
  Case ID: 7  Line: 52   Col: 4   Filename: example-kcov.c
  Case ID: 8  Line: 55   Col: 4   Filename: example-kcov.c
  ?: ID: 9   Line: 56   Col: 9   Filename: example-kcov.c
  Default ID: 10 Line: 59  Col: 4   Filename: example-kcov.c
  If ID: 11  Line: 64   Col: 2   Filename: example-kcov.c
  ?: ID: 12  Line: 64   Col: 7   Filename: example-kcov.c
  ImpDef. ID: 13 Line: 68  Col: 2   Filename: example-kcov.c
  Case ID: 14 Line: 69   Col: 3   Filename: example-kcov.c
  Case ID: 15 Line: 72   Col: 3   Filename: example-kcov.c
  Do ID: 16  Line: 77   Col: 2   Filename: example-kcov.c
  If ID: 17  Line: 77   Col: 2   Filename: example-kcov.c
Total number of branches: 30
```

Note 1. We count each case as one branch (i.e., considering `switch() {...}` has multiple outgoing edges). Also, we count (implicit) default statement as one branch regardless of whether default exists or not. A line and a column of an implicit default is those of corresponding `switch()`.

2. (10 pts) Print out the branches in the attached `grep` source code file (i.e., `grep.c`) by using your `kcov-branch-identify`. Submit the output of your `kcov-branch-identify` on `grep.c`.

Note 1. If your program fails to find header files of a target program (`grep.c`), you have to modify `include_paths` (line 137) in the initialization part of the `kcov-branch-identify.cpp` template file.

Note 2. You can ignore various Clang warnings.

Note 3. Your program should print out functions which have no branches.

Note 4. The total # of branches of the `grep` C file: > 3000

3. (90 pts) Write down `kcov` using Clang. You have to submit your `kcov` code.
- `kcov` receives a file name of a *preprocessed* single C file `<f>.i` (which is generated from non-preprocessed C file `<f>.c` using the below command) and generates the instrumented version `<f>-cov.c` to measure branch coverage of `<f>.c` through testing.
 - A preprocessed C file can be obtained by `gcc -E <filename>.c -o <filename>.i`
 - Note. If you give a complex C file like `grep.c` w/o **preprocessing** to `kcov` as an input, `kcov` may crash due to the high complexity of handling source code location by Clang Rewriter.
 - When `<f>-cov.c` is compiled and executed 1st time, `<f>-cov.c` generates a coverage measurement file `coverage.dat`. After then, `<f>-cov.c` updates `coverage.dat` through testing `<f>-cov.c`. The format of `coverage.dat` is as follows

```

Line#|# of execution |# of execution | conditional
      |of then branch |of else branch | expression
1453      0          0          errnum
1474      0          7          size && !result
1484      3          0          ptr
1488      0          0          size && !result
...
6950      0          0          (end = memchr(beg + len, '\n', (buf + size) -
(beg + len))) != 0
6955      0          0          beg > buf && beg[-1] != '\n'
Covered: 581 / Total: 3101 = 18.735892%

```

Note1. If one line has multiple branches (i.e., nested `if` statements), you can print out these branches in separate lines with the same line id

Note2. The # of execution of else branch of case should be always 0 (i.e., meaningless)

Note3. For a `switch` statement, your program should print out case and (implicit) default statements. A conditional expression of case statement is a corresponding case value and that of default is `default`

4. (10 pts) Print out the coverage measurement file of the *preprocessed* `grep` C code with the following test cases (execution commands) where `grep.c` is the `grep` source code file used for your HW (not preprocessed C file)

```

./grep -n "if" grep.c
./grep -E "[0-9][0-9]+" grep.c
./grep -E "[[:digit:]][[:alpha:]]" grep.c

```