ENEE 457 Static Analysis Class Exercise

- 1. Assume we have an analyzer that takes as input any C program and has the following properties:
 - a) The analyzer always terminates
 - b) If the C program makes an array out-of-bounds memory access during its run on an input x, then the analyzer outputs 1.
 - c) If the C program does not make an array out-of-bounds memory access during its run on an input x, then the analyzer outputs 0.

Show that the analyzer can be used to solve the Halting Problem.

ENEE 457 Static Analysis Class Exercise

2. Consider the following code snippet on which we would like to perform a taint analysis. Type qualifiers are represented by capital letters: A, B, C, D, E.

```
1
     int printf(A char *fmt, ..);
     B char *fgets(..);
\mathbf{2}
3
4
\mathbf{5}
     int main () {
               C char *mystring = fgets(.., network_fd);
6
\overline{7}
               D char *mystring2 = mystring;
               E char *mystring3 = ''Hello World'';
8
               mystring2 = mystring3;
9
               printf(mystring2);
10
               return 0;
11
     }
12
```

- i. Identify all the sources and sinks in the code snippet and determine the corresponding settings for the type qualifiers.
- ii. List all of the constraints on the type qualifiers.
- iii. Is there a vulnerability in the above code? Is there a solution for the undetermined type qualifiers that satisfies all the constraints? If there is no vulnerability and no solution, it means that our taint analysis has produced a false positive. How can the taint analysis be modified so that the false positive is removed?