**Malicious Adversary**

In this project, you will play the role of an evil, malicious adversary whose goal in life is to make programs run slowly. A malicious adversary, such as yourself, has the ability to look into the algorithm or a program to develop the worst-case input for that program.

You will find three programs named prog1.cpp, prog2.cpp and prog3.cpp. Please first download them to your working directory.

Each of these accepts input from standard input in the same format: an integer , followed by non-negative integers. The maximum value of is 100,000. Your goal is to carefully examine these three programs, and to design inputs for them that will cause them to run slowly. Moreover, you are to write three programs bad1.cpp, bad2.cpp, and bad3.cpp, that each respectively generate bad inputs for prog1.cpp, prog2.cpp and prog3.cpp. Each of your programs should take a single argument on the command line giving the input size, and it should print to standard output a bad input case of that size. To test your code, you could for example run the following:

g++ prog1.cpp -o prog1

g++ bad1.cpp -o bad1

./bad1 100000 > input1

./prog1 < input1

For prog3, you may also want to re-direct the output to a file, since there is a large amount of output, so that the time spent printing the output does not obscure the overall running time:

./prog3 < input1 > output1

Ideally, you should make each of the sample programs take 5 seconds or more for large input cases; the exact running time will depend on the machine you use, of course. Your programs bad1.cpp and bad2.cpp should run very quickly, in time. It is ok if your program bad3.cpp runs in time, as long as you explain in a comment how you could make it run in time (fully implementing the version takes perhaps a bit more time than we have for this assignment, sadly...)

You are not allowed to change prog1.cpp, prog2.cpp, or prog3.cpp

Please create a compressed archive containing files bad1.cpp, bad2.cpp, and bad3.cpp:

tar -zcvf Malicious\_Adversary.tar.gz bad1.cpp bad2.cpp bad3.cpp