

---

**The University of Queensland**  
**School of Information Technology and Electrical Engineering**  
**Semester One, 2009**

**COMP2303 / COMP7306 – Assignment 2**

**Due: 11pm Monday April 20, 2009**

**Marks: 50**

**Weighting: 25% of your overall assignment mark (COMP2303)**

## **Introduction**

The goal of this assignment is to ensure you have gained familiarity and skills with both the C programming language and using a debugger (such as gdb(1)) to examine various characteristics of running programs. These will be essential skills in later assignments for this course.

For this assignment you will be given an executable program (the “bomb”) which you have to “defuse” by entering the correct defusing phrases into the program for each of the 10 bomb phases.

This is an individual assignment. You should work on defusing your own bomb by yourself. You should feel free to discuss aspects of C programming and the use of debuggers with your fellow students, but you shouldn’t actively help (or seek help from) other students with the defusing of particular phases. You should note that each student will receive a different bomb and the strings that defuse your bomb will be different to the strings that defuse another student’s bomb.

## **Obtaining the “Bomb”**

Whilst logged in to agave, the student UNIX server, you should type the following command:

```
~comp2303/a2/getbomb
```

This will create a subdirectory within your current directory named `comp2303a2` and place the bomb files into that directory. The files will include an executable called *bomb* and a number of source files (.h and .c files). Your bomb (executable and source) will be different to the bombs for all other students. You will not receive all of the source files – just some of them.

There is enough information contained within the bomb executable and the supplied source files in order for you to successfully defuse all phases (although some of them are more difficult than others). You should note that some of the bomb’s modules have been compiled with debugging support (-g flag to gcc) and some haven’t.

## **Running the “Bomb”**

The *bomb* program will only run on agave and you are the only user who can run your *bomb* program. Any attempt to run the program on another host or to run another user’s bomb will cause the bomb to exit immediately.

Whilst in your `comp2303a2` directory, you can execute the bomb by typing

```
./bomb
```

**You may not want to do this until you are ready to try defusing the bomb.** When you start the bomb program, it will print out details of any phases you have already defused and it will print your current mark (out of 50) and the maximum mark you can obtain based on your attempts to date.

---

The bomb will then prompt you to enter the string which defuses the first phase you haven't yet defused. If you enter an incorrect string, the bomb will "explode" and exit. You can then run the bomb program again to try again. You will lose marks for every time the bomb "explodes".

You may temporarily pass on a phase by entering the string

PASS

(This can be upper or lower case, but must have no surrounding spaces etc.) You will then be prompted to enter the defusing string for the next phase and may return to the passed phase later.

Once you successfully defuse a phase, you will never have to defuse that phase again.

You may exit the bomb program by entering the string

QUIT

(This can be upper or lower case, but must have no surrounding spaces etc.)

You should note that the bomb is booby trapped. You are warned against modifying the internal data structures of the bomb – you never know what might happen.

## Hints

There is a TEST phase that does not count for marks. You may access this phase by entering the string

TEST

(This may be upper or lower case, but must have no surrounding spaces etc.) You will then be prompted to enter the defusing string for the test phase. Once you have entered the TEST phase, you can not PASS on it and must either complete it correctly or fail trying (in which case the bomb explodes but does not cost you any marks).

You should carefully read the supplied source code and be familiar with the use of gdb before attempting to run the bomb. If you run the bomb and type in an unexpected phrase, it will explode and you will lose marks. It is suggested you run the bomb from within a debugger rather than standalone.

All phases have associated code and some debugging information and you will need to use a debugger to set breakpoints, examine various variables etc in order to determine the defusing strings. You may need to use a number of features of gdb including watchpoints, automatic display, conditional breakpoints, breakpoint command lists.

You should note that the code which determines each defusing string is not executed until AFTER the defusing text is read from the user so you may need to enter some arbitrary text, debug the code to determine the defusing string, quit the program and then run it again to enter the defusing string for that phase.

The function which determines whether the string input by the user matches the secret string for each phase (and causes the explosion if there is no match) is called `secret_string_matches()`. It prompts the user to confirm their attempt (by typing 'ok') before continuing. You may want to type

```
break secret_string_matches
```

into gdb to ensure that you don't enter this function unless you really want to.

---

## Submission

Every time you run the bomb, a record is kept of your interactions with it and your success/failure at defusing each phase. Your submission time for the assignment will be considered to be the time of your last attempt to defuse any phase of the bomb. You must make at least one attempt in order to be considered to have made a submission. An attempt means either that the bomb explodes or a phase is defused.

The due date for this assignment is 11pm Monday April 20, 2009. The policy on “grace days” as outlined in the course profile is applicable. Submissions completed before 11pm Sunday April 19, 2009 will earn additional grace days. Note that no submissions can be made more than 120 hours past the deadline (i.e. after 11pm Saturday April 25, 2009) under any circumstances. (The bomb program will continue to function for a short time after this, but any interactions you have with it will be discounted when computing your final mark.)

## Marks

There are 10 phases, each worth 5 marks. The mark you achieve for each phase is determined by the number of attempts taken before you successfully defuse that phase. If you defuse a phase on the first attempt, you will receive 5 marks for that phase. If it takes you longer than one attempt, your mark for that phase will be

$$5 \times 0.8^{(\text{number\_of\_attempts}-1)}$$

i.e. if it takes you 2 attempts, your mark for that phase will be 4 out of 5, 3 attempts gives you 3.2 out of 5, 4 attempts gives you 2.56 out of 5, etc. There is no limit on the number of attempts you can make at any phase before succeeding. You should note that although each phase is worth the same number of marks, they are not of equal difficulty.

Late penalties will apply as outlined in the course profile. All marks are subject to an audit of our logs to ensure that you have correctly entered the defusing strings and haven't tampered with the bomb to defuse it in some other way.