Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.087: Practical Programming in C

IAP 2010

## Problem Set 1

Writing, compiling, and debugging programs. Preprocessor macros. C file structure. Variables.
Functions and program statements. Returning from functions.

**Out:** Monday, January 11, 2010. **Due:** Tuesday, January 12, 2010.

## Problem 1.1

(a) What do curly braces denote in C? Why does it make sense to use curly braces to surround
the body of a function?

(b) Describe the difference between the literal values 7, "7", and '7'.

(c) Consider the statement
**double** ans = 10.0+2.0/3.0−2.0∗2.0;

Rewrite this statement, inserting parentheses to ensure that ans = 11.0 upon evaluation of
this statement.

## Problem 1.2

Consider the statement
**double** ans = 18.0/squared(2+1);

For each of the four versions of the function macro `squared()` below, write the corresponding value
of `ans`.

1. `#define squared(x) x*x`

2. `#define squared(x) (x*x)`

3. `#define squared(x) (x)*(x)`

4. `#define squared(x) ((x)*(x))`

## Problem 1.3

Write the "Hello, 6.087 students" program described in lecture in your favorite text editor and
compile and execute it. Turn in a printout or screen shot showing

- the command used to compile your program

- the command used to execute your program (using `gdb`)

- the output of your program

## Problem 1.4

The following lines of code, when arranged in the proper sequence, output the simple message "All your base are belong to us."

1. `return` 0;

2. `const char` msg[] = MSG1;

3. }

4. `#define` MSG1 `"All your base are belong to us!"`

5. `int` main(`void`) {

6. `#include` <stdio.h>

7. puts(msg);

Write out the proper arrangement (line numbers are sufficient) of this code.

## Problem 1.5

For each of the following statements, explain why it is not correct, and fix it.

(a) `#include <stdio.h>;`

(b)
```
int function(void arg1)
{
   return arg1-1;
}
```

(c)
```
#define MESSAGE = "Happy new year!"
puts(MESSAGE);
```

6.087 Practical Programming in C
January (IAP) 2010