

Homework 3

1. What's the output of the following C program?

```
#include <stdio.h>
#include <string.h>

int main()
{
    int x = 6;
    char *s = "hel\0 w0rld!";
    int y = strlen(s);
    int z = (y - x) % 2;
    printf("%d %d %d\n", y, z, s[11]);
    return 0;
}
```

2. Integer representation using 1's and 2's complement. Assume we have an integer type of 8 bits, fill in the following table.

Value	2's complement	1's complement	Sign-Magnitude
27	0001 1011	0001 1011	0001 1011
	0110 1000		
		1101 0101	
			1011 0101
-127			

3. Will the following “bomb” boom? Why?

```
#include <stdio.h>

unsigned i;

int main()
{
    for (i=10; i>=0; i--) {
        sleep(1);
        printf("%s\n", (i&1)?"tock":"tick");
    }
    printf("BooM!\n");
    *((int *)0) = 0xdeadbeef;
}
```

4. Consider the following C expressions:

```
short s = -5;
unsigned short us = s;
int i = -121;
unsigned int ui = i;
```

Assume we are running code on an 8-bit machine using 2's complement for signed

integers.

Also assume that right shifts of signed values are performed arithmetically. A “short” integer is encoded using 4 bits. Fill in the blanks in the table below.

Expression	Binary Representation
(int)17	0001 0001
(int) -17	
us	
ui	
(i << 1) - 15	
(s >> 2) << 1	
(us >> 2) << 1	
(i >> 2) - (ui >> 2)	

5.

In C language, if an evaluation expression contains both unsigned and signed values, then signed values will be implicitly casted into unsigned ones before evaluation. Please fill the following table with “<”, “>” or “==”. (Assume int value is encoded using 16 bits)

Constant A	Relation	Constant B
-1	>	0U
0		-1
-3U		-2U
-32768		32768U
-1		65535U
-32768		-1U