CMSC 216 Worksheet #2 Fall 2012

1. Fill in the following table with the numbers' equivalents in different bases. The first row is provided for you.

Binary	Octal	Hexadecimal	Decimal
0001 0010	022	12	18
11001	031	0x19	25
101100	054	2C	44
111111	077	0x3F	63
0011 0011	063	0x33	51

2. Give the output of the following program:

```
#include <stdio.h>

int main() {
  unsigned char a = 0xa5, b = 0x1c, c = 0xf0;

  printf("a & b: %02x\n", a & b);
  printf("a & c: %02x\n", a & c);
  printf("a | c: %02x\n", a | c);
  printf("b ^ c: %02x\n", b ^ c);
  c = ~c;
  printf("~c: %02x\n", c);

  return 0;
}
```

a & b: 04

a & c: a0

a | c:f5

b ^ c: ec

~c: 0f

3. Give the output of the following program:

```
#include <stdio.h>

int main() {
  unsigned char a = 25, b;

  while (b)
    b--;

  while (a) {

    if (a & 0x80)
      b++;

    a <<= 1;

    if (a)
      b <<= 1;

    printf("%d\n", b);
  }

  return 0;
}
```

0

0

0

2

6

12

24

25

4. Give the output of the following program:

```
#include <stdio.h>

int main() {
  unsigned int a = 1, b = 9;
  while (a++ - b--)
    printf("%d\n", a - b);
  return 0;
}
```

-6

-4

-2

5. The following program is spread across two files; give its complete output. You may assume the program is compiled via the command "gcc -o prog file1.c file2.c" and run by executing "./prog".

file1.c

```
#include <stdio.h>

extern int a;
static int b;

void f(int);
void g(void);

int main() {
  a = 10;
  b = 20;
  f(a);
  f(b);
  g();
  printf("main: %d %d\n", a, b);

  return 0;
}
```

file2.c

```
#include <stdio.h>

int a;
static int b;

void f(int c) {
  static int b = 5;

  a += b;
  b += c;
  printf("f: %d %d\n", a, b);
}

void g(void) {
  a += 5;
  b = 10;
  printf("g: %d %d\n", a, b);
}
```

f: 15 15 {a is 10, b init to 5, 10+5 5+ 10}

f: 30 35 {a is 15 b is 0 until noticed again when it becomes 15}

g: 35 10 {a is global, so all of the previous changes are still accessible b is 10}

main: 35 20 35 b here is 20….