Oklahoma State University
ENSC 3213 - Computer-based Systems - Laboratories
Spring 2019
Homework 01: Introduction to bitwise masking and C

SOLUTION

1. (2.5 points) Convert the following binary numbers into decimal values:

0000 0100 0110 1001 -> 1,129

0011 0001 0111 1111 -> 12,671

0101 0101 0101 0101 -> 21,845

2. (2.5 points) Convert the following hexadecimal numbers into decimal values:

1A -> 26

FF -> 255

21 -> 33

3. (2.5 points) Represent the following in two's complement form using 16 bits:

-29 -> 1111 1111 1110 0011

165 -> 0000 0000 1010 0101

-100 -> 1111 1111 1001 1100

4. (2.5 points) Suppose the initial value of X is 0b1111, which mask and bitwise operation should be used to make X = 0b1010? **Hint:** You should answer using a couple of lines of C code!

```
mask = 0b0101;

X = X & ~(mask); // Bitwise clear
```

5. (2.5 points) Suppose the initial value of `X` is `0b0000`, which mask and bitwise operation should be used to make `X = 0b1010`? **Hint:** You should answer using a couple of lines of C code!

```
mask = 0b1010;

X = X | mask; // Bitwise set
```

6. (2.5 points) Suppose `X` is a four bit value, and its initial value is **unknown** (`X = 0b????`), which mask and bitwise operation should be used to make `X = 0b1010`? **Hint:** You should answer using a couple of lines of C code! In this case, you will need to use two different masks and perform two different bitwise operations.

```
mask1 = 0b0101;

X = X & ~(mask1); // Bitwise clear

mask2 = 0b1010;

X = X | mask2; // Bitwise set
```

7. (2.5 points) Suppose `Mask = 0x00000F0F` and `P = 0xABCDABCD`. What are the results of the following bitwise operations?

```
(1) Q = P & Mask;
Q = 0x00000B0D
(2) Q = P | Mask;
Q = 0xABCDAFCF
(3) Q = P ^ Mask;
Q = 0xABCDA4C2
(4) Q = ~Mask;
Q = 0xFFFFF0F0
(5) Q = P & ~Mask;
Q = 0xABCDA0C0
```

8. (2.5 points) Given the following variables and their initializations:

```
int a, x, y, z;
float b, u, v;

x = 10; y = 20; z = 30;
u = 4.0; v = 10.0;
```

What are the values of the expressions in each of the following problems:

```
(1) a = x - y - z;
a = -40
(2) a = x + y * z;
a = 610
(3) a = z / y + y;
a = 21
(4) a = x / y / z;
a = 0
(5) b = v - u;
b = 6.0
(6) b = v / u;
b = 2.5
```

9. (2.5 points) Under what conditions are the following expressions True?

`(1) (x == y && y == z)`

Only if `x = y = z`.

`(2) (x == y || y == z)`

Only one of these three possibilities are true:

(a) $x = y = z$

(b) $x = y \neq z$

(c) $x \neq y = z$

`(3) (x > y && x < z)`

Only if `x > y` and `x < z` at the same time.

10. (2.5 points) We wish to print integers from 1 through 10. Check if the following loop will do so correctly.

```
i = 1;
while (i < 10) {
        printf("\%d\n", i);
        i = i + 1;
}
```

No. The code snippet contains an error.
In C, the type of all variables must be declared. So, the first line should be `int i = 1;`.
The while loop itself will only print integers from 1 through 9.
The correct code would be:

```
int i = 1;
while (i <= 10) {
        printf("\%d\n", i);
        i = i + 1;
}
```