CSC 209 Review 9 Solution

September 13, 2020

1. a) 8

   Correct Solution

   0

   b) 0

   c) 1

   d) 6

   Correct Solution

   15

   Notes

   - `a)` is 0 because

     $i >> 1 + j >> 1$ is $8 >> 10 >> 1$

     Which is $0 >> 1$

     Which is 0.
   - `d)` is 15 because

     $ij\&k$ is $78\&9$

     which is 78

     which is 15

2. Use XOR on target bit using value 1.

   This is because the operator of two like values equals to 0, and unequal values equal to 1.

a	b	a ^ b
0	0	0
0	1	1
1	0	1
1	1	0

3. The macro switches the value of `x` and `y`.

   Take for example $x = 100$ (8) and $y = 010$ (4)

   For the first part of macro, we have $x = x\text{^}y = 100\text{^}010 = 110$.

   Taking this to second part of macro, we have $y = y\text{^}x = 010\text{^}110 = 100$.

   Lastly, we have $x = x\text{^}y = 110\text{^}100 = 010$.

   Thus, we can see the value of x and y are switched.

4.
```
#define MK_COLOR(red,green,blue) (long) ((blue << 16) | (blue | (
green << 8)) | red)
```

5.
```
#define GET_RED(color) (int)(color & 255)

#define GET_GREEN(color) (int)((color >> 8) & 255)

#define GET_BLUE(color) (int)((color >> 16) & 255)
```

6. a) Please see file `question_6_a.c` for details.
   b) Please see file `question_6_b.c` for details.

   Notes

   - Unisigned short has at max 4 bits.
   - Any out-of-bound bits are omitted

7. Please see file `question_7.c` for details.

8. a) Returns first `n` bits of 1
   b) Extracts `n` bits from `m-n+2`th bit

9. a) Please see file `question_9_a.c` for details.

   b) Please see file `question_9_b.c` for details.

10. Please see file `question_10.c` for details.

11. The precedence of &, ^, and | is lower than the equality operators.

    So, given `if (key_code & (SHIFT_BIT | CTRL_BIT | ALT_BIT) == 0)`, (SHIFT_BIT | CTRL_BIT | ALT_BIT) == 0) will be evaluated first, which is incorrect.

    To fix this problem, add parenthesis to `key_code & (SHIFT_BIT | CTRL_BIT | ALT_BIT)`.

12. The precedence of + is higher than <<. So, `8 + low_byte` in `high_byte << 8 + low_byte` will be evaluated before `high_byte <<`.

    To fix this problem, add parenthesis to `high_byte << 8`.

13. All bits in `n` are gradually reduced to 0, starting from the right-most bit.

14.
```
union ieee_float {
    float value;
    struct {
        unsigned int fraction: 23;
        unsigned int exponent: 8;
        unsigned int sign: 1;
    } parts;
};
```

Correct Solution

```
struct ieee_float {
    unsigned int fraction: 23;
    unsigned int exponent: 8;
    unsigned int sign: 1;
};
```

15. a) This is because the value of `int`'s sign in some compiler is oppsite to the others (e.g. 0 represents a positive sign in some compiler, where as 1 represents a positivie sign in other compilers).

    b) To avoid this problem, use `unsinged int` instead.

16.
```
typedef unsigned long DWORD;
typedef unsigned short WORD;
typedef unsigned char BYTE;

union {
    struct {
```

```
7             DWORD eax, ebx, ecx, edx;
8         } dword;
9         struct {
10            WORD axl, axh, bxl, bxh, cxl, cxh, dxl, dxh;
11        } word;
12        struct {
13            BYTE al, ah, ale, ahe, bl, bh, ble, bhe, cl, ch, cle, che,
    dl, dh, dle, dhe;
14        } byte;
15    } regs;
```