

CMSC 216 Quiz 3 Worksheet

The third quiz for the course will be on Wed, Oct 9. The following list provides additional information about the quiz:

- The quiz will be a written quiz (no computer).
- The quiz will be in lab session.
- Closed book, closed notes quiz.
- Answers must be neat and legible.
- Quiz instructions can be found at <http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html>
- Make sure you know your section number and your TA's name.

The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours. It is recommended that you try these exercises on paper first (without using the computer).

Exercises

1. Convert **45₁₀** to binary and hexadecimal.
2. Convert **01101101₂** to decimal and hexadecimal.
3. Assuming the variables **a** and **b** have the following values:

a → 01101101₂ and **b → 10101001₂**

What is the result of the following operations (provide your answers using hexadecimal numbers).

a & b → a | b → a ^ b → ~b → a << 4 → b >> 7 →

4. Write the 2's complement representation of -17.
5. What is the output of the following program?

```
#include <stdio.h>

static void process(int *f) {
    printf("V6: %d\n", f[0]);
    printf("V7: %d\n", f[1]);
}

int main() {
    int data[] = {10, 7, 30, 4};
    int *p = data;
    int *m = data + 2;

    printf("V1: %ld\n", p - m);
    printf("V2: %ld\n", m - p);
    printf("V3: %d\n", p[1]);
    p++;
    printf("V4: %d\n", p[1]);
    *(data + 1) = 9;
    printf("V5: %d\n", p[-1]);
    printf("V6: %d\n", data[1]);

    process(data + 1);

    return 0;
}
```

6. Implement a function called **get_values** that has the prototype:

```
void get_values(unsigned int value, unsigned int *p_lsb, unsigned int *p_msb,
               unsigned int *p_combined)
```

The function computes the least significant byte and the most significant byte of the **value** parameter. In addition, both values are combined. For this problem:

- You may not use any loop constructs.
- You may not use the multiplication operator (`*` or `*=`).
- Your code must work for unsigned integers of any size (4 bytes, 8 bytes, etc.).
- To combine the values, append the least significant byte to the most significant one.
- Your implementation should be efficient.

The following driver (and associated output) provides an example of using the function you are expected to write. Notice that in this example an unsigned int is 4 bytes, but your function needs to work with an unsigned int of any size.

Driver

```
int main() {
    unsigned int value = 0xabcdfaec, lsb, msb, combined;
    get_values(value, &lsb, &msb, &combined);
    printf("Value: %x, lsb: %x, msb: %x, combined: %x\n", value, lsb, msb, combined);

    return 0;
}
```

Output

Value: abcdfaec, lsb: ec, msb: ab, combined: abec

```
void get_values(unsigned int value, unsigned int *p_lsb, unsigned int *p_msb,
               unsigned int *p_combined)
```

7. What is the size (`sizeof(How_Many_Bytes)`) of the following structure, given the following machine parameters:

`sizeof(char) == 1; sizeof(int) == 4; sizeof(long) == 8; sizeof(char *) == 8;`

Integer values must be aligned.

```
typedef struct how_many_bytes {
    long s;
    char c, e;
    int i;
    char *d;
} How_Many_Bytes;
```

8. Given two `How_Many_Bytes` structures `p1` and `p2`, which of the following operations are legal?

- `p1 == p2`
- `How_Many_Bytes.i = 10;`
- `How_Many_Bytes *p1_ptr = &p1;`
- `int *x = &p1.i;`

Would assigning `p1` to `p2` (`p2 = p1`) make `p2` a complete copy (deep copy) of `p1`?