

CMSC 216 Quiz 2 Worksheet

The second quiz for the course will be on Tue, Jun 18 (in lab). The following list provides additional information about the quiz:

- The quiz will be a written quiz (no computer).
- 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.
2. Convert **01101101₂** to decimal.
3. Convert the previous binary number to hexadecimal.
4. Assuming the variables **a** and **b** have the following values:

$a \rightarrow 01101101_2$ and $b \rightarrow 10101001_2$

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

- $a \& b \rightarrow$
- $a | b \rightarrow$
- $a \wedge b \rightarrow$
- $\sim b \rightarrow$
- $a \ll 4 \rightarrow$
- $b \gg 7 \rightarrow$

- Write the 2's complement representation of -17.
- Write a function that has as prototype **void printBits(unsigned int val)**. The function prints the parameter value in bit format. For example, the function call **printBits(22)** will print (for 4-byte unsigned integer):

[illegible]

For this problem:

- You cannot assume all unsigned integers are 4-bytes. Use `sizeof()` to determine the number of bytes associated with an unsigned integer.
 - Your solution must be efficient.
 - You may not use arrays.
7. What is the initial value of the following variable pointer?

```
int *m_ptr;
```

8. Which of the following pointer variables uses the largest number of bytes?

```
int *p;
char *q;
```

9. What will happen when the following code is executed? Explain briefly.

```
int *p = NULL;
*p = 20;
```

10. Is the following code valid? Explain briefly.

```
int *p;
p = 0;
```

11. Write a code fragment that can potentially generate a segmentation fault.

12. Why it is not possible to dereference a void pointer?

13. Why is the following function wrong?

```
int *get_value(int x) {
    int y = x * 2;

    return &y;
}
```

14. Define a function `max_min` that compute the maximum and minimum for two values. The function prototype is: **`void max_min(int x, int y, int *max, int *min);`**

15. Draw a memory map for the following function that shows the values of the variables before the return 0 statement is executed. In cases where you are asked to print an address, write **NULL** or **MEMORY_ADDRESS** (for any other value).

```
int main() {
    int age = 41, capacity = 5;
    int *age_ptr = &age;
    int *p, *r;
    int **q = &p;

    p = age_ptr;
    r = &capacity;
    printf("%d\n", *age_ptr);
    printf("%p\n", age_ptr);
    printf("%d\n", *p);
    printf("%d\n", **q);
    printf("%d\n", *r);
    q = &r;
    printf("%d\n", **q);

    return 0;
}
```

16. To the right of the provided code, write the output generated by the program below. In cases where you are asked to print an address, write **NULL** or **MEMORY_ADDRESS** (for any other value).

```
void first(int *p) {
    *p += 3000;
    p = NULL;
}

void second(int **p) {
    **p -= 3;
    *p = NULL;
}

int main() {
    int cds = 10;
    int dvds = 77;
    int *q = &cds;
    int *m = &dvds;
    int **r = &q;

    printf("V1: %d\n", *m);
    printf("V2: %d\n", **r);
    cds += 100;
    printf("V3: %d\n", *q);
    printf("V4: %d\n", *m);
    first(q);
    printf("V5: %d\n", *q);
    printf("V6: %p\n", q);
    second(&m);
    printf("V7: %p\n", m);
    printf("V8: %d\n", dvds);

    return 0;
}
```