

CMSC 216 Quiz 2 Worksheet

The second quiz for the course will be on Wed, Sep 25. 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.
- **You need to be familiar with the following string library functions: strcmp, strlen, strcpy**

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. What makes a character array a string? Is every character array a string?
2. Which of the following pointer variables uses the largest number of bytes?

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

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

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

4. Is the following code valid? Explain briefly.

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

5. Is the following code valid? Explain briefly.

```
int x = 20;  
void *p = &x;  
p = p + 1;
```

6. Define a function called `is_palindrome`. The function will return 1 if the string parameter is a palindrome and 0 otherwise.
7. The function `filter` has the following prototype:

```
static int filter(const int src[], int dest[], int array_size,  
                 int lower_bound, int upper_bound)
```

The function initializes the `dest` array with elements that have values in the range defined by `lower_bound` (inclusive) and `upper_bound` (inclusive). The size of `src` and `dest` is `array_size`. The function will return the number of elements placed in `dest`. You can assume `lower_bound` is less than or equal to `upper_bound`.

8. Implement the function **maximum** that has the prototype below. The function computes the maximum in the array and returns that value via the max parameter. If the array has a size of 0, the pointer variable associated with the argument must be set to NULL. The following code fragment illustrates how the function will be used.

```
int b[] = {30, 5, 80, 4};
int max;
int * max_ptr = &max;

maximum(b, 4, &max_ptr);
if (max_ptr == NULL) {
    printf("Array size is 0\n");
} else {
    printf("%d\n", max);
}
```

You can assume the array passed to the function will have positive elements (if the array size is different from 0).

```
static void maximum(const int a[], int a_size, int **max)
```

9. Define a method call intersection that has the following prototype:

```
static int intersection(const int *a, int size_of_a, const int *b, size_of_b,
                        int *common);
```

The method will initialize the common array with elements that are present in arrays a and b. The method will return the number of elements in common. You can assume the common array is large enough to fit the result.

10. Define a method that takes as parameter an array of strings and returns the longest string in the array. Use the string library.