

CMSC 216 Quiz 3 Worksheet

The next quiz for the course will be on Mon, Feb 16. The following list provides additional information about the quiz:

- Do not post any solutions to this worksheet in Piazza. That represents an academic integrity violation.
- 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). **At the end we have provided an example of a memory map, so you know exactly what we are expecting while drawing maps. Take a look at the example before drawing any maps.**

Exercises

1. You need to be familiar with the debugging guide available at:
<http://www.cs.umd.edu/class/spring2015/cmsc216/content/resources/debugging.html>
2. What is the difference between a pointer and a pointer variable?
3. What is a NULL pointer?
4. Which of the following pointer variables occupies the largest number of bytes?

```
int *x;  
float *y;  
double *m;
```

5. Why do we need to specify the type of a pointer variable?
6. How many pointer variables can be associated with a particular variable?
7. When will a segmentation fault occur when we dereference NULL? For example,

```
int *ptr = NULL;  
printf("%d\n", *ptr);
```
8. How are pointer arguments passed in C? By value?
9. What is the output of the following program? Would it be possible to get a segmentation fault?

```
#include <stdio.h>  
  
int main() {  
    int *ptr;  
  
    *ptr = 400;  
    printf("%d\n", *ptr);  
  
    return 0;  
}
```

10. Write a code fragment that shows that NULL is considered false in C.

11. What does the name of an array represent?
12. When do you want to use the const modifier?
13. Could you assign NULL to the name of the array?
14. What are the values associated with the following array?

```
int a[5] = {2, 8, -3};
```

15. Draw a memory map for the following program up to the point indicated by the comment **/*HERE */**. In addition, provide the output generated by the program.

```
#include <stdio.h>

#define MAX_LEN 5

static void task(int *b, int range) {
    b[range - 1] = 200;
    range = 0;
    b = NULL;
    /* HERE */
}

int main() {
    int a[] = {2, 4, 6};
    int len = 3, i;

    task(a, len);
    printf("len %d\n", len);
    for (i = 0; i < len; i++) {
        printf("%d\n", a[i]);
    }

    return 0;
}
```

16. Define a function that receives two integer arrays as parameters and determines whether the arrays have the same elements. For this function:
 - a. The function returns 0 if the arrays are different and 1 otherwise.
 - b. The arrays may have different lengths.
 - c. Feel free to add any parameter you might need.
17. Define a function that receives an integer array as parameter and shifts all the array elements one position to the left (the first element will become the last one). You may not define/create a new array in the function.

Sample Memory Map

We are providing this example so you know what we are expecting for memory maps.

Example

Draw a memory map for the following program up to the point indicated by the comment **/*HERE */**.

```
#include <stdio.h>

#define MAX_LEN 5

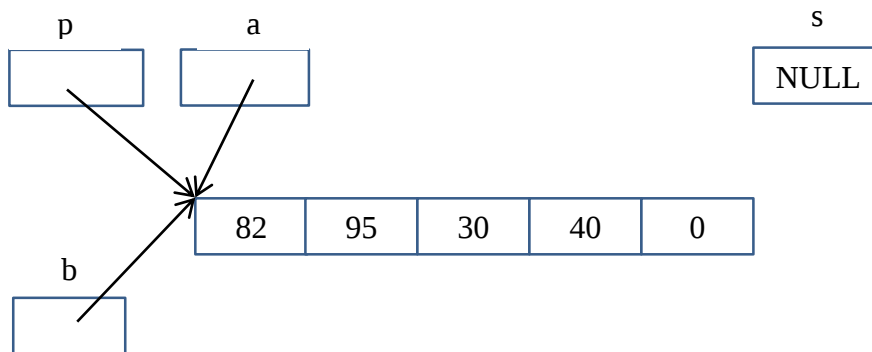
void process(int *b, int *s) {
    b[0] = 82;
    s[1] = 95;
    s = NULL;
    /* HERE */
}

int main() {
    int a[MAX_LEN] = {10, 7, 30, 40};
    int *p = a;

    process(a, p);

    return 0;
}
```

Answer:



Note: You can also replace NULL with the ground symbol as done in lecture. For example, **s** above could be represented as:

