

COMP 166 – Week 9 Quiz

Name: _____

1. Write a code fragment to declare an integer array, `arr`, and fill it with the first 1000 *odd* integers. That is, fill it with the values 1, 3, 5, 7, ... , until you have 1000 values. Declare any variables you use. You do not have to write a whole program; just write the lines that accomplish the above task.

2. Write a code fragment that prints out the *address* that a pointer, `p`, points to, and the *value* that is stored at that referenced address. Assume that the value is a `double` type. Therefore, the first value you print out will be an address and the second value you print out will be a `double`.

NOTE: assume the pointer is already declared and initialized – you don't have to do that part. You also don't know what the address and value are; you are to just write the `printf()` statements that would display them.

COMP 166 – Week 9 Quiz

3. Fill in the Symbol and RAM tables for the following code fragment:

```
int x = 5;
int y = 10;
int * p = &y;
p = p+2;
```

NOTE: assume the compiler starts memory allocations at address 1000, and subsequent allocations go towards higher address numbers. You may also assume that integers and addresses take 4 bytes each.

| Symbol Table | | | RAM | |
|--------------|------|---------|---------|-------|
| Type | Name | Address | Address | Value |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

4. Write the *prototype* for a function called `count` that has three parameters: a 1-D array of integers, an integer representing the size of the array, and an integer target value. *Use pointer notation to define the array parameter!*

NOTE: you are not to write the body of the function, just the prototype!

5. Consider the following 2-D array declaration:

```
int m[7][3];
```

What is the 1-dimensional *offset* of the element `m[4][2]`? Offset: _____