### POINTER REVIEW

1. **int *a;**  
   *a = 12;  
   ? Creates memory for the storage of an integer?  
   **ANS: NO**  
   ? What is the value of a?  
   **ANS: uninitialized; no way to know where a is being stored.**

2. **int a = 12;**  
   **int *d = &a;**  
   *d = 10 - *d;  
   **d = 10 - *d;**  
   **What is in a?**  
   **OK? YES... *d = a → a = 10 - a**  
   **OK? NO... ptr_var ← int − int ILLEGAL**

3. ***&a = 25**  
   **Assuming int a;**  
   **OK? What does it do? a = 25, but way too much overhead.**

4. ***100 = 25**  
   **OK? NO... can only use indirection (*) on expressions of type pointer.**

5. **Store 25 at address 100**  
   **(int *) 100 = 25**  
   **What? If ever need to access the HW itself**

6. **How is c declared?**  
   **int a = 12;**  
   **int *b = &a;**  
   **c = &b;**  
   **a = ?**  
   **b = ?**  
   ***b = ?**  
   **c = ?**  
   **c = ?**  
   ***c = ?**  
   **int **c = &b;**  
   **var ptr ptr**  
   **a 12 b &a c &b**  
   **a = 12**  
   **b = &a**  
   ***b = a, 12... *b = *&a (see #3)**  
   **c = &b**  
   **c = b, &a... *c = *&b = b = &a → 12**  
   ****c = *b, a, 12**

7. **char *string;**  
   **int length = 0;**  
   **while (*string++ != \’\0\’) length += 1;**  
   **If string is initialized correctly, what is length?**  
   **The number of characters until the NUL byte is reached (strlen function)**  
   **PROBLEM??? Empty string OK but not NULL ptr... should check for before enter while loop.**

8. **#define N_VALUES 5**  
   **float values[N_VALUES];**  
   **float *vp;**  
   **for (vp = &values[0];**  
   **vp < &values[N_VALUES];)**  
   ***vp++**  
   **Draw:**