**Problem 1**

1. True
   1. SIZE is defined as 4, and i will only ever increase as long as it is less than SIZE. Therefore, foo1 will always return 3.
2. False
   1. The initial value of ptr is the address of num
3. False
   1. To exit, continue1 must equal 0, which happens when \*ptr equals 3
4. False
   1. It’s the same as scanf(“%d”,&num);
5. False
   1. The return value of foo is an inputted number + 4
6. True
   1. ptr is pointing to num, so if ptr is being dereferenced, then they are the same variable.
7. T?
   1. The program would continue to run, but it would output “Not a valid entry.”
   2. Although it is seen as true, 3 is the only value of input that will equal in quitting the program.
8. True
   1. foo has an int pointer, and foo1 has a char pointer array. Both have 1 parameter
9. True
   1. ptr and &num are the same value, as ptr is the pointer of num
10. False
    1. In foo, a dereference operator is used in the return statement.
11. True
    1. Both parameters are 8 bytes
12. False
    1. In foo1, since the print statement is utilizing the %s format specifier, it will output the string(char array) beginning at the address listed.
13. False
    1. w[i] holds an address

**Problem 2**

1. Creating and initializing a variable (by hardcoding)
2. Creating and initializing a variable (by hardcoding)
3. Creating and initializing a variable (by hardcoding)
4. Assigning a value to a variable (by storing a function return value/using a function)
5. Assigning a value to a variable (by storing a function return value/using a function)
6. Assigning a value to a variable (by storing a function return value/using a function)
7. Assigning a value to a variable (by hardcoding)