Danny Tan

1. The program will compile unless it modifies some important data which might lead to a crash in the program.
2. The base address of the array will be passed.
3. It will print 3, 2, 15

x = ++a[1] will increment the value of a[1] first and then set it to x which will be 2

y= a[1] ++ will set y = a[1] which is 2 before the increment.

z = a[x++] will access z = a[x] first which before it increment x which will set z = a[2] = 15. Then it will increment x by 1 which will turn it into 3. Then it prints x,y,z which will be 3,2,15.

1. One advantage for garbage collection is that programmers don’t have to worry about freeing memory allocated which can lead to less memory leaks.  The advantage of having no garbage collection is that the programmer can manage the memory and he/she knows when the memory is being freed or data is getting deleted.
2. &x = 502

y = 502

\*y = 502

&y = error

\*(\*y) = error

1. No error because a pointer is always 8 bytes in a 64-bit system so the compiler will know the size of the structure which will be (8+4) 12 bytes.
2. There will be an error because you cannot fit the same structure inside itself. Also, the complier does not know the size of the structure when the program is complied.
3. a. 8

b. 8

1. Pointer only takes up a constant amount of memory allocation, 8 bytes in a 64-bit system so this will save up a lot of space in memory. A pointer allows us to access a variable that is defined outside of the function.
2. a. The variable will be deleted after the function is finished therefore it will be out of scope so the user won’t have access to the variable anymore.

b. Similar to part a, the pointer variable will be deleted when it is out of scope and this will cause a memory leak since we cannot access the memory we allocated anymore.

1. a. Because we have to account for 0 in the range, it can only go up to 2n – 1. For example, if we have 22, we can only go up to 0 to 3.

b. Because 0 is included in the positive range. So there will be 2n-1 negative numbers, 0, and 2n-1 – 1 positive numbers.

c. More negative because 0 is accounted for in the positive range and not in negative range, so there are more negative numbers than positive numbers.

1. a. 170

b. -86

c. AA

d. No because it is just a shorthand of writing binary. We convert 4 binary digits into one hexadecimal digit.