Hyun Ho Kim 1001967176

1.True, filename.txt and 3 have to be switched because argv[2] has to be a filename, and argv[1] has to be a number in this code.

2.False, it depends on the number of argv[2], so we cannot do not know how much structs it would be.

3.False, we need increase the index to replace it by adding des++; at the end of the loop.

4.True, we allocate memories by multiplying the number of the argument, which is on the second command line, and the size of structure.

5.True, des has its own address, so declaring double pointer would not make any errors.

6.False, char \*flavor in the structure is just a pointer. We need some spaces to type information in.

7.True, we allocate des’s bytes by multiplying the second argument on the command line not the first argument and the size of the structure.

8.True, it passes the size of bytes of the structure dessert to the function malloc so it will allocate more bytes as many as the num value multiplies.

9.True, it passes the number which indicates how many structures it will make, and the for loop will not stop until it makes that many of structures. Eventually, it will be same as the sizeof(des).

10.True, it passes an address of structures in address, so we can use either one.

11.False, des is a struct pointer, so it has 8 bytes. However, dessert is not a pointer.

12.True, so it can move to the next contiguous memory address.

13.False, argv[2] is an address of a second index of char \*\*. Double pointers cannot hold the address.

14.True, because (\*d).flavor++; increase one index.