1 Describe and explain what happens when you try to compile a program with the following statements: (1)

```
int N = 1000;
int[] a = new int[N*N*N*N];
```

2 Consider the following, two-dimensional array of integers: (1)

```
int[] numbers = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
```

What value is given by `numbers[1][2]`?

3 Write the method, `reverse()`, that reverses the order of a one-dimensional array of integers. (2)

4 Write the method, `scalarMult()`, that accepts a two-dimensional array of `doubles` and a `double` scalar and returns a two-dimensional array of the same dimensions in which each element is the original value multiplied by the given scalar. (2)
Note: Your method should not modify the original array passed to it.

5 *Euclidean Distance.* The *Euclidean Distance* between two points of arbitrary dimension, $(a_1, b_1, c_1, ...)$ and $(a_2, b_2, c_2, ...)$ is defined as: (4)

$$d = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2 + \cdots}$$

For example, the distance between points $(1, 4, 3, 2)$ and $(2, 6, 7, 8)$ can be computed as:

$$\begin{aligned} d &= \sqrt{(1-2)^2 + (4-6)^2 + (3-7)^2 + (2-8)^2} \\ &= \sqrt{1 + 4 + 16 + 36} \\ &= \sqrt{57} \\ &\approx 7.5498 \end{aligned}$$

Write a method that will accept two arrays of doubles representing n-dimensional points and calculate the *Euclidean Distance* between them.
Note: Treat as a precondition that each array is of equal length.

6 (6)

(a) *Longest Sequence.* Given an array of integers, find the length and location of the longest contiguous sequence of equal values.

(b) *Longest Plateau.* Given an array of integers, find the length and location of the longest contiguous sequence of equal values where the values of the elements just before and just after the sequence are smaller.

Note: Only submit your solution to part (b) of this question.