Assignment 1: Java Programming

1. **(5 points)** The class `Progression` has a single field, named `current`. It defines two constructors, one accepting an arbitrary starting value for the progression and the other using 0 as the default value. The remainder of the class includes three methods:

   - `nextValue()`: A public method that returns the next value of the progression, implicitly advancing the value each time.
   - `advance()`: A protected method that is responsible for advancing the value of current in the progression.
   - `printProgression(int n)`: A public utility that advances the progression n times while displaying each value.

   Write a Java class `AbsoluteProgression` that extends the `Progression` class so that each value in the progression is the absolute value of the difference between the previous two values. You should include

   - `AbsoluteProgression(long first, long second)`: A parametric constructor that starts with a specified pair of numbers as the first two values (2 points).
   - `AbsoluteProgression()`: A default constructor that starts with 2 and 200 as the first two values (2 points).
   - A `main` method to print such a progression of length 5 (1 point).

   (Note: It will be 0 point if the new class `AbsoluteProgression` does not extend the provided class `Progression`.)

2. **(10 points)** Refer to the given class `SinglyLinkedList`, which implements partially a singly linked list (not all of the required operations). Complete the following methods for this class (2.5 points each):

   - `SinglyLinkedList(E[] elements)`: A constructor that builds a singly linked list from an array of a generic type[1].

[1] Updated from the assignment posted earlier. You can also follow the earlier version.

- `printElements()`: Print the elements in the list from `head` to `tail`.
- `reverse()`: Reverse the singly linked list by using only a constant amount of additional space.
- `main()`: A testing method that displays the elements in the list and those after the list is reversed.

3. **(10 points)** Write a short **recursive** Java method that rearranges an array of integer values so that all the even values appear before all the odd values. Use the following array to test and print out the rearranged array:

   ```
   int[] data = 1, 4, 7, 2, 10, 5;
   ```

   (Hint: Consider whether the last element is odd or even and then put it at the appropriate location based on this and recur.)

   (Note: It will be 0 point for a non-recursive method.)