In Class Exam 1. NAME:____________________________________ (There are THREE 3 Questions)

DO NOT WRITE the class `TextBook` but suppose there is a class `TextBook` that has fields for title, authors, publisher, ISBN and also includes a field `Pagination`. We are going to write the class `Pagination` and provide some functionality.

All classes are in the package `exam01`.

Write your code on the sheets provided.

Question 1. Make a class `Pagination`. It has 4 private fields: `numPages` (the number of pages in the book) and `numChapters` (the numbers of chapters in the book) are `ints`. Then there are two arrays of `int` called `chapStarts` and `chapAssigned`. The value of `chapStarts[n]` will be the page number where Chapter *n* starts. The value of `chapAssigned[n]` will be 0 or 1. If it is 1 then Chapter *n* has been assigned for the students to read by the instructor. To avoid confusion, we will leave `chapStarts[0]` and `chapAssigned[0]` with the value 0, since there never is a Chapter 0.

The constructor of `Pagination` has 2 `int` parameters, one for `numPages` and one for `numChapters`. The constructor also instantiates the two arrays as `int` arrays of length `numChapters+1`. Note that making the array length 1 bigger means that the last element in the array corresponds to the number of the last chapter in the book, which is `numChapters`. *Remember the elements of a new* `int` *array are* 0 *by default.*

Write the method `public void setChapStartPage(int n, int p)`, which sets the starting page of Chapter *n* to *p*, that is, it gives `chapStarts[n]` the value `p`.

Write the method `public void assign(int n)`, which sets the element for Chapter *n* to 1, that is, it gives `chapAssigned[n]` the value 1, indicating that it is part of the reading for the course.

The hardest part of the question is the next method:

Write a method `public int totalToRead()`, which returns the total number of pages that have been assigned to read. You can assume that the last page number of any chapter is the page immediately before the start page of the next chapter (for example if Chapter 5 starts on page 100, then Chapter 4 ends on page 99). You have to treat the last chapter differently if it is part of the readings assigned. Assume the last chapter ends on the last page (which is the field `numPages`)—we will ignore the index at the end of the book. HINT do not forget you would also have to count the last page of the book if the last chapter was assigned to the readings!

In the class `Pagination` write a main method that creates a `Pagination` object with 4 chapters and `numPages` equal to 158, the chapters starting on pages 1, 55, 89, and 128 and only Chapters 2 and 4 assigned as reading. Then print out the expected value and the actual value you get by calling `totalToRead`.

Question 2. Look at this class and what the test program produces and then write the code of the method `modify` on the separate sheets provided. The Javadoc of the method explains how it works.

```
package exam01;

import java.util.ArrayList;
import java.util.Arrays;

public class Exercise {
      private int pos;
      private char ch;

      public Exercise(int pos, char ch) {
            if(pos < 0) throw new IllegalArgumentException("the int cannot be negative");
            this.pos = pos;
            this.ch = ch;
      }

      /**
       * Does nothing if list is null. The int pos and the char ch are the fields above.
       * If list.size() is less than or equal to pos, then
       * it adds the char ch to list until list.size() equals pos + 1. Otherwise,
       * if list.size() is greater than pos, it changes the char at index pos and
       * ALL following char in list to the char ch.
       * @param list the ArrayList<Character> to be modified to have the char ch from
       * index pos onwards
       */
      public void modify(ArrayList<Character> list) {

            //WRITE THE CODE OF THIS METHOD ON THE BLANK SHEETS PROVIDED

      }

      public static void main(String[] args) {
            Exercise e = new Exercise(5, '+');
            ArrayList<Character> list =
                        new ArrayList<>(Arrays.asList('a','b','c','d','e','f','g'));
            System.out.print(list + " => ");
            e.modify(list);
            System.out.println(list);
            list = new ArrayList<>(Arrays.asList('a','b','c','d','e','f'));
            System.out.print(list + " => ");
            e.modify(list);
            System.out.println(list);
            list = new ArrayList<>(Arrays.asList('a','b','c','d','e'));
            System.out.print(list + " => ");
            e.modify(list);
            System.out.println(list);
            list = new ArrayList<>(Arrays.asList('a','b','c','d'));
            System.out.print(list + " => ");
            e.modify(list);
            System.out.println(list);
            list = new ArrayList<>(Arrays.asList('a','b','c'));
            System.out.print(list + " => ");
            e.modify(list);
            System.out.println(list);
            list = new ArrayList<>();
            System.out.print(list + " => ");
            e.modify(list);
            System.out.println(list);
            list = null;
            System.out.print(list + " => ");
            e.modify(list);
            System.out.println(list);
      }
}
```

OUTPUT:

```
[a, b, c, d, e, f, g] => [a, b, c, d, e, +, +]
[a, b, c, d, e, f] => [a, b, c, d, e, +]
[a, b, c, d, e] => [a, b, c, d, e, +]
[a, b, c, d] => [a, b, c, d, +, +]
[a, b, c] => [a, b, c, +, +, +]
[] => [+, +, +, +, +, +]
null => null
```

Question 3 (about using “super” in a constructor and a method). Write the code for the class TweetQuestion as a subclass of Question, so that running the main method of QuestionTest gives the output shown:

```
package exam01;

public class Question {
      private String question;

      public Question(String text) {
            question = text;
      }

      public void display() {
            System.out.println("Write your answer to the question below:");
            System.out.println(question);
      }
}


public class QuestionTest {
      public static void main(String[] args) {
            Question q1 = new Question("What did you do on your last vacation?");
            Question q2 = new TweetQuestion("Which course are you enjoying most?");
            q1.display();
            System.out.println("-------------------");
            q2.display();
      }
}
```

OUTPUT:

```
Write your answer to the question below:
What did you do on your last vacation?
-------------------
You must limit your answer to 140 characters.
Write your answer to the question below:
Which course are you enjoying most?
```

Write your code for the class TweetQuestion as a subclass of Question here: