Practice Lab Assignment 3

Practice Lab Assignment 3

For this Practice Lab Assignment, you will write basic programs to practice Java syntax.

Instructions

- There are 5 questions in this assignment.
- In case of any technical glitch, discuss with TA.

Due Date: Show your codes to TA to get the attendance and Submit your remaining codes on BB.

Submission Format: A zip file containing all .java files.

Grading Criteria

No Grading Criteria.

Questions

1. Generate an array of n integers. **Write recursive functions** to do the following tasks:

    (a) the smallest even number in the array,
    (b) the largest odd number in the array and
    (c) the largest prime factor among the factors of all elements in the array.

    You can use the below template for part (a) and (b) and accordingly write a function for (c).

```
import java.util.*;
public class Main
{
   public static int largestOdd(int[] arr, int n, int index)
  {

   }

 public static int smallestEven(int[] arr, int n, int index)
  {
   }
```

```
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++)
            arr[i] = s.nextInt();
        int even = smallestEven(arr, n, 0);
        System.out.println("Smallest Even Number in the Array is:" +even);
        System.out.println("\n");
        int odd = largestOdd(arr, n, 0);
        System.out.println("Largest Odd Number in the Array is:" +odd);
    }
}
```

2. Create an array of size 11. Print the array elements. Write a program to reverse an array, without using any additional array. Print the reversed array.

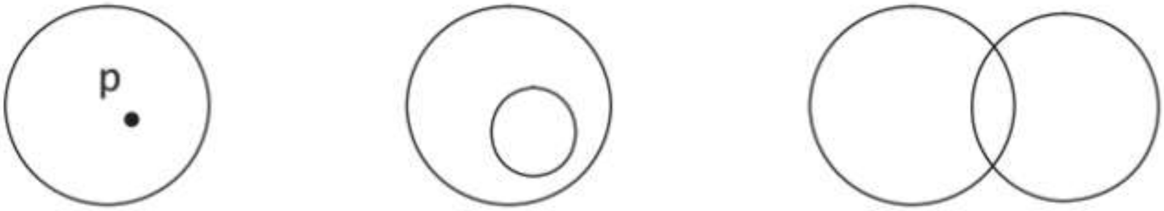
3. Write a program to sort a list of values in ascending order. The user might input a String array or an Integer array. Depending on the type of input, you need to call the respective version of **Void sort()** method. There are three variations of the method. Use any sorting method (Bubble, Selection or Insertion).

    - **Void sort(int[] arr, int n)** - expects an integer array ‘arr’, and n denotes the size of the array
    - **Void sort(int[] arr, int n, boolean reverse)** - expects an integer array ‘arr’, and n denotes the size of the array. It also has an additional `reverse` parameter which when set to **true** returns the array sorted in descending order.
    - **Void sort(String[] arr, int n)** - expects a String array ‘arr’, and n denotes the size of the array. In this case, the array needs to be sorted lexicographically.

4. Write a Program to design a class having static member function named show_count() which has the property of displaying the number of objects created of the class.

5. Define the Circle2D class that contains:
    - Two double data fields named **x** and **y** that specify the center of the circle with getter methods.
    - A data field **radius** with a getter method.
    - A parameterless constructor that creates a default circle with (0, 0) for (x, y) and 1 for radius.
    - A constructor that creates a circle with the specified x, y, and radius.
    - A method **getArea()** that returns the area of the circle.

- A method **getPerimeter()** that returns the perimeter of the circle.
- A method **contains(double x, double y)** that returns true if the specified point (x, y) is inside this circle (see Figure a).
- A method **contains(Circle2D circle)** that returns true if the specified circle is inside this circle (see Figure b).
- A method **overlaps(Circle2D circle)** that returns true if the specified circle overlaps with this circle (see Figure c).



(a) (b) (c)

Write a test program that creates a Circle2D object c1 (new Circle2D(2, 2, 5.5)), displays its area and perimeter, and displays the result of c1.contains(3, 3), c1.contains(new Circle2D(4, 5, 10.5)), and c1.overlaps(new Circle2D(3, 5, 2.3)).