Programming Assignment (#3)
Introduction to Java
Points: 50

Instructions:

- Submit the following:
  YourASURiteID-Assignment3.zip This compressed folder should contain the following files:
  1. PalindromicPrimes.java
  2. EmployeeWorkHours.java
  3. A readme.txt containing any instructions you want to provide to the instructor/grader
- Make sure your java files compile without any compiler errors. You will not receive any credit for programs with compiler errors.
- If you are unable to complete your program, submit the parts that work with no compiler errors for partial credit.

Problem 1: Printing Palindromic Primes 25 points

A *palindromic prime* is a prime number and also palindromic. For example, **131** is a prime and also a palindromic prime. Write a program that that takes in an integer n as input and displays the first n palindromic prime numbers. Display **10** numbers per line and align the numbers properly as shown below. Make sure the display and alignment looks good for upto *n=100*.

Make use of methods to write this program.

Definition of Prime Number: A Prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.

File name: PalindromicPrimes.java

Sample Output:

```
    2     3     5     7    11   101   131   151   181   191

  313   353   373   383   727   757   787   797   919   929
```

. . . . .

Problem 2: Computing the weekly hours of Employees

25 points

Suppose the weekly hours for all employees are stored in a two-dimensional array. Each row records an employee's n-day work hours with n columns where $n \geq 1$ and $n \leq 7$ representing the number of days in a week that these employees work. For example, the table shown below represents an array that stores the work hours of eight employees for 7 days in a week. Write a program that takes in as inputs, the number of employees and the number of working days in a week. Then it takes in all the Employee information (name and number of daily hours worked). This program should display employees and their total hours worked in a week in decreasing order of the total hours.

File name: EmployeeWorkHours.java

Sample Input Data:

	Sun	*Mon*	*Tue*	*Wed*	*Thur*	*Fri*	*Sat*
Employee 0	2	4	3	4	5	8	8
Employee 1	7	3	4	3	3	4	4
Employee 2	7	3	4	3	3	2	2
Employee 3	9	3	4	7	3	4	1
Employee 4	3	5	4	3	6	3	8
Employee 5	3	4	4	6	3	4	4
Employee 6	3	7	4	8	3	8	4
Employee 7	6	3	3	9	2	7	9

Sample Output:

```
Employee 7 worked 39 hours
Employee 6 worked 37 hours
Employee 0 worked 34 hours
Employee 4 worked 32 hours
Employee 3 worked 31 hours
Employee 1 worked 28 hours
Employee 5 worked 28 hours
Employee 2 worked 24 hours
```