# Introduction to Programming

Labs – Week 5

## Exercise 1

Write a program `allDigitsOdd.java` that determines whether every digit of a given positive integer is odd. Print true if the number consists entirely of odd digits (1, 3, 5, 7, 9) and false if any of its digits are even (0, 2, 4, 6, 8). For example,

```
> java allDigitsOdd 135319
true
```

```
> java allDigitsOdd 9145293
false
```

## Exercise 2

Write a program called `gcd.java` that accepts two integers as parameters and returns the greatest common divisor (GCD) of the two numbers. The GCD of two integers `a` and `b` is the largest integer that is a factor of both `a` and `b`.

One efficient way to compute the GCD of two numbers is to use Euclid's algorithm, which states the following:

```
GCD(a,b) = GCD (b, a % b)
GCD(a,0) = absolute value of a
```

## Exercise 3

Write a program `CountEven.java` that create array of `int` from input and print the count of even integers in the array.

Hint: To read value from command-line, create an `int` array of size `args.length` and convert the integers given as `String` from `args` array. To read from `Scanner`, first ask user the length `n` of the array, and then in a loop read n value using `sc.nextInt()`.

## Exercise 4

Write a program `isSorted.java` that given an array of real numbers prints `true` if the list is in sorted (non-decreasing) order and `false` otherwise. For example, if arrays store `{16.1, 12.3, 22.2, 14.4}` and `{1.5, 4.3, 7.0, 19.5, 25.1, 46.2}` respectively, your program should print `false` and `true` respectively. Assume the array has at least one element. A one-element array is considered to be sorted.

Note: You may use hard coded array in this exercise.

## Exercise 5

Write a program `ReverseArray.java` that reverses the order of values in a one-dimensional `String` array. For instance, the following array `{"zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"};` would be transformed `to {"nine", "eight", "seven", "six", "five", "four", "three", "two", "one", "zero"}`

Do not create another array to hold the result. Use exchanges/swap to transform the array.

Note: You may use hard-coded array in this exercise.

## Exercise 6

Write a program `MaxOnes.java` that given an array of integers, find the maximum number of consecutive `1`'s present in the array.

*Example*

```
{1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0} → 4
{0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1} → 1
```

Note: You may use hard-coded array in this exercise.

## Exercise 7

*Birthday problem.* Suppose that people enter an empty room until a pair of people share a birthday. On average, how many people will have to enter before there is a match? Write a program `Birthday.java` to simulate one experiment. Write a program `Birthdays.java` to repeat the experiment many times and estimate the average value. Assume birthdays to be uniform random integers between 0 and 364.

Hint: Use an array of `boolean` values to mark the birthdays of people present in the room.

## Exercise 8

*(Day of the week)* Write a program `DayOfWeek.java` that takes a date as input and prints the day of the week that date falls on. Your program should take three command-line arguments: `m` (month), `d` (day), and `y` (year). For `m` use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar (where / denotes integer division):

$$y_0 = y - (14 - m) / 12$$

$$x = y_0 + y_0/4 - y_0/100 + y_0/400$$

$$m_0 = m + 12 * ((14 - m) / 12) - 2$$

$$d_0 = (d + x + 31m_0 / 12) \; mod \; 7$$

For example, on which day of the week was August 2, 1953?

$$y_0 = 1953 - 0 = 1953, \quad x = 1953 + 1953/4 - 1953/100 + 1953/400 = 2426$$

$$m_0 = 8 + 12 * 0 - 2 = 6, \quad d_0 = (2 + 2426 + (31 * 6)/12) \; mod \; 7 = 2443 \; mod \; 7 = 0 \text{ (Sunday)}$$