

# CS 101 - Algorithms & Programming I

Fall 2022 - Lab 6

Due: Week of November 21, 2022

Remember the honor code for your programming assignments.

For all labs, your solutions must conform to the CS101 style guidelines!

All data and results should be stored in variables (or constants where appropriate) with meaningful names.

The objective of this lab is to learn static methods. Remember that analyzing your problems and designing them on a piece of paper *before* starting implementation/coding is always a best practice.

For the methods below you should **not** use built-in methods that perform the mentioned tasks directly. The mentioned methods should be implemented from scratch.

## 0. Setup Workspace

Start VSC and open the previously created workspace named `labs_ws`. Now, under the `labs` folder, create a new folder named `lab6`.

In this lab, you are to have one Java class/file (under `labs/lab6` folder) as described below. A second Java file containing the revision should go under this folder as well. We expect you to submit a total of 2 files including the revision. Do *not* upload other/previous lab solutions in your submission. The user inputs in the sample runs are shown in `blue` color.

## 1. Password decoder

First, we will design and implement some *static* methods. Then these methods will be used in a program to decode a password input by the user.

For security purposes, the passwords entered by users in an online store are encoded and then sent to a server. We want to implement the decoder of the password that will be used on the server. A password is a number of  $N$  digits where each number is decoded using the following function:

$$d(x) = \text{floor}(10 - (x/2 - 3)^2) - 0.32$$

Where the floor is the largest integer smaller than the given float number. You will write a program that will get an  $N$  digit integer and return the decoded version as an integer.

*Implement helper method:*

- `int floor(double)`: should return the largest integer smaller than the input

*Implement helper method (use the Floor helper method):*

- `int digitDecoder(int)`: given an integer  $x$ , should calculate  $d(x)$  as mentioned above

*Use helper method in:*

- `int passwordDecoder(int)`: returns the decoded password for the password given as input

Sample runs:

```
Please input the password: 237619082
The decoded password is: 579937085
```

## 2. Prime Element Summation

A number can be written as the multiplication of its prime factors. For instance  $126 = 2 \times 3^2 \times 7$ . You are going to write a program that returns the sum of the prime factors of a given number. In the case of 126, the output should be  $2 + 3 + 7 = 12$ .

*Implement helper method:*

- `boolean isPrime(int)`: returns whether or not the given integer is prime

*Use helper method in:*

- `int primeSummation(int)`: returns the sum of prime elements of a positive integer

Sample runs:

```
Please input the natural number: 210
The summation of the prime factors of 210 is 17.
```

```
Please input the natural number: 89
The summation of the prime factors of 89 is 89.
```

## 3. Pythagorean Checker

A **Pythagorean triplet** is a set of three numbers where the square of one number is equal to the sum of the square of the other two. In other words, for the set of three numbers, a, b, and c:  $a^2 = b^2 + c^2$ . Please note a, b, and c can be ordered in different ways and all cases should be checked.

*Implement helper method:*

- `int power(int, int)`: this is your own implementation of `Math.pow(r,n)`

*Use helper methods in:*

- `boolean isPythagorean(int, int, int)`: returns whether or not the given numbers form a Pythagorean triplet

Sample runs:

```
Please input the first number: 5
Please input the second number: 4
Please input the third number: 3
The triplet is Pythagorean, power(5, 2) = power(3 ,2) + power(4 ,2)
```

```
Please input the first number: 5
Please input the second number: 7
Please input the third number: 3
The triplet is not Pythagorean.
```

## 4. Binary AND/OR Operations

In this question, you will write a program that gets two positive integers (int's) from the user and performs binary bitwise AND and OR operations on them (again represented as int's). You will also need to implement a helper converter method that converts a number to its binary representation (represented as a long containing 0's and 1's as digits) and use it in the binary operation methods. For this purpose, you will need to implement three static methods:

*Implement helper method:*

- `long intToBinary(int)`: should return integer form of the binary representation of the input

*Use helper method in:*

- `long binaryAND(int, int)`: returns the binary AND of two inputs

*Use helper method in:*

- `long binaryOR(int, int)`: returns the binary OR of two inputs

**Note:** You may assume that the binary representation of the entered integers will fit in the `long` data type.

Sample runs:

```
Please enter the first natural number: 105
Please enter the second natural number: 91
Binary representation of the first number: 1101001
Binary representation of the first number: 1011011
The bitwise AND operation result: 1001001
The bitwise OR operation result: 1111011
```