**Due date:** Today, at the end of the lab period.

**Read this entire document before beginning your lab.**

The *Comp248LabManual* is available on the course's Moodle website.

For this lab you are **required to fulfill all requirements exactly as described** in this provided document, no less, no more.

**Question**: Today you are commissioned to write a Java program that will prompt for and read 3 integer values as *coffee, burger, and fries* that are entered by the user and decides whether the order consists of two types of combos, i.e., trio combo and burger combo. *One trio combo contains one burger, one fries and one coffee. One burger combo contains one burger and one fries*. The program first tries to decide whether there are any trio combos based on the order items, then whether there are enough **remaining** order items to form burger combos. Note that an order may belong to more than one of the categories below. Be sure to use the same format and wording as in the sample runs in the table below.

1. If the order contains 0 coffee, 0 burger or 0 fries, then there is no way to form a trio combo, display "No trio combos from this order!".
2. If the order contains 0 coffee, but does contain some burgers and some fries, display "Only burger combos from this order!".
3. If the order contains 0 burger or 0 fries, then there is no way to form burger combo, so display "No burger combos from this order!".
4. For the orders with at least one item from each category, both trio combo and burger combo may be found, if there are still remaining burgers and fries after forming trio combos as many as possible, display "This order contains some trio combos and some burger combos!", otherwise display "Only trio combos from this order!".
5. The box below illustrates how your program should behave and appear. REMEMBER in the output: ◦ is a space and ↵ is a new line. Text in green is user input

```
How◦many◦cups◦of◦coffee?◦100↵
How◦many◦burgers?◦50↵
How◦many◦fries?◦50↵
↵
Only◦trio◦combos◦from◦this◦order!
```

```
How◦many◦cups◦of◦coffee?◦0↵
How◦many◦burgers?◦50↵
How◦many◦fries?◦50↵
↵
No◦trio◦combos◦from◦this◦order!↵
Only◦burger◦combos◦from◦this◦order!
```

```
How◦many◦cups◦of◦coffee?◦1↵
How◦many◦burgers?◦2↵
How◦many◦fries?◦0↵
↵
No◦trio◦combos◦from◦this◦order!↵
No◦burger◦combos◦from◦this◦order!
```

```
How◦many◦cups◦of◦coffee?◦1↵
How◦many◦burgers?◦2↵
How◦many◦fries?◦2↵
↵
This◦order◦contains◦some◦trio◦combos◦and◦some◦
burger◦combos!
```

```
How◦many◦cups◦of◦coffee?◦1↵
How◦many◦burgers?◦0↵
How◦many◦fries?◦1↵
↵
No◦trio◦combos◦from◦this◦order!↵
No◦burger◦combos◦from◦this◦order
```

**Note 1:** You are to expect a perfect user who will always enter 3 non-negative integers; that is, **do not** verify the validity of user input.

**<u>Note 2:</u>** The use of libraries other than *java.util.Scanner* is prohibited. Your program must work for any non-negative integers entered, not just the ones in the samples above.

**<u>Note 3:</u>** Final thought, remember that your solution is case-sensitive and space-sensitive and fulfill the above instructions carefully and precisely.

**<u>Reminder</u>:**

When submitting your solution to the lab system, make sure that  if you have a `package` statement at the top of your `.java` file it is commented out (has // in front of it) as failing to do so will result in a grade of 0 (restriction of the DOMJudge system).