ASSIGNMENT 1

COMP-202, Fall 2017, Section 001-002-003

Due: September **28th**, 2017 (23:59)

Please read the entire PDF before starting. You must do this assignment individually.

Question 1:	45 points
Question 2:	25 points
Question 3:	30 points
	100 points total

It is very important that you follow the directions as closely as possible. The directions, while perhaps tedious, are designed to make it as easy as possible for the TAs to mark the assignments by letting them run your assignment, in some cases through automated tests. While these tests will never be used to determine your entire grade, they speed up the process significantly, which allows the TAs to provide better feedback and not waste time on administrative details. Plus, if the TA is in a good mood while he or she is grading, then that increases the chance of them giving out partial marks. :)

To get full marks, you must:

- Follow all directions below
- Make sure that your code compiles
  - Non-compiling code will receive a very low mark
- **Write your name and student number in a comment in each .java file you hand in**
- Indent your code properly
- Name your variables appropriately
  - The purpose of each variable should be obvious from the name
- Comment your work
  - A comment every line is not needed, but there should be enough comments to fully understand your program

Part 1 (0 points): Warm-up

Do **NOT** *submit this part, as it will not be graded. However, doing these exercises might help you to do the second part of the assignment, which will be graded. If you have difficulties with the questions of Part 1, then we suggest that you consult the TAs during their office hours; they can help you and work with you through the warm-up questions. You are responsible for knowing all of the material in these questions.*

Warm-up Question 1 (0 points)
Create a file called `HelloWorld.java`, and in this file, declare a class called `HelloWorld`. This class should define only one method called `main()`. In the body of this method, use `System.out.println()` to display "`Hello world!`". You can find such a class in the lecture slides; make sure you can compile it and run it properly.

Warm-up Question 2 (0 points)
Create a file called `Diagram.java`, and in this file, declare a class called `Diagram`. This class should define only one method called `main()`. In the body of this method, use five statements of `System.out.println()` to display the following pattern:

```
  22
2   2
   2
 2
22222
```

Warm-up Question 3 (0 points)
Practice with Binary:
We usually use base 10 in our daily lives, because we have ten fingers. When operating in base 10, numbers have a `one`s column, a `ten`s column, a `100`s column, etc. These are all the powers of 10.

There is nothing special about 10 though. This can in fact be done with any number. In base 2, we have each column representing (from right to left) 1,2,4,8,16,etc. In base 3, it would be 1,3,9,27, etc.

Answer the following short questions about number representation and counting.

1. In base 10, what is the largest digit that you can put in each column? What about base 2? Base 3? Base n?
2. Represent the number thirteen in base 5.
3. Represent the number thirteen in base 2.
4. What binary number is equal to the sum of these two binary numbers? 10101011 + 10010001
5. What is the number from the previous part in base 10?
6. What is the binary number for 11010010 + 11000101?
7. And what is the number from the previous part in base 10?

Warm-up Question 4 (0 points)
Answer the following questions.

1. What does the following logical expression evaluate to?
   `(false or false) and (true and (not false))`
2. Let a and b be boolean variables. Is it possible to set values for a and b to have the following expression evaluate as *false*? `b or (((not a) or (not a)) or (a or (not b)))`

Warm-up Question 5 (0 points)
The following code is designed to put the value of the variable x into the variable y and the value of the variable y into x. What is wrong with it?

```
public class BadSwap {
     public static void main(String[] args) {
          int x = 1;
          int y = 2;
          x = y;
          y = x;
     }
}
```

Part 2

The questions in this part of the assignment will be graded.

Question 1: ISBN (45 points)

The reference book suggested for Comp202 is 'How to Think Like a Computer Scientist: Java Version, 6th edition, 1491929561'. The last 10-digit number in the book (i.e., 1491929561) is the ISBN of the book. ISBN is the acronym for **I**nternational **S**tandard **B**ook **N**umber and it identify a specific book. One particularity of this number is such that $10d_{10}+9d_9+8d_8+\cdots+3d_3+2d_2+d_1$ must be a multiple of 11 (please note that d_i denotes the i^{th} digit from the right). For example, for the ISBN of our reference book we can check the condition as follows: $(10\times1+9\times4+8\times9+7\times1+6\times9+5\times2+4\times9+3\times5+2\times6+1 = 253$, and 253 is multiple of 11). By convention, the digit d_1 can be any value from 0 to 10 (where the character 'X' is used to denote de digit '10'). The other digits (from d_{10} to d_2) must all be digits between 0 and 9. For example, given the 9-digit ISBN number 020131452 the value of d_1 is equal to 5 since is the only value of d_1 between 0 and 10 for which $10\times0+9\times2+8\times0+7\times1+6\times3+5\times1+4\times4+3\times5+2\times2+d_1$ is a multiple of 11. For this assignment, we will work with a shorter version of an ISBN number (a number with only five digits). Particularly, as a Comp202 student, you have been commissioned to write a java program to determine the digit d_1 given the other 4 digits (i.e., d_5, d_4, d_3 and d_2). You can assume that the input number will not have any leading 0s.

Let's see three examples:

SAMPLE INPUT 1:
2956
SAMPLE OUTPUT 1:
4

NOTES (for INPUT 1):
i) The input represents the ISBN number 2956?, where $d_5 = 2$, $d_4 = 9$, $d_3 = 5$, $d_2 = 6$ and $d_1 =?$ (the value that your algorithm has to compute).
ii) The output of your problem should be the digit 4. Please note that 4 is the only value of d_1 between 0 and 10 for which $5 \times 2 + 4 \times 9 + 3 \times 5 + 2 \times 6 + 4 = 77$ is a multiple of 11.

SAMPLE INPUT 2:
5724
SAMPLE OUTPUT 2:
X

NOTES (for INPUT 2):
i) The input represents the ISBN number 5724?, where $d_5 = 5$, $d_4 = 7$, $d_3 = 2$, $d_2 = 4$ and $d_1 =?$ (the value that your algorithm has to compute).
ii) The output of your problem should be the character X. Please note that 10 is the only value of d_1 between 0 and 10 for which $5 \times 5 + 4 \times 7 + 3 \times 2 + 2 \times 4 + 10 = 77$ is a multiple of 11. In this case you must print the letter 'X' because by convention the character 'X' must be used to denote de digit '10'.

SAMPLE INPUT 3:
1521
SAMPLE OUTPUT 3:
0

NOTES (for INPUT 3):
i) The input represents the ISBN number 1521?, where $d_5 = 1$, $d_4 = 5$, $d_3 = 2$, $d_2 = 1$ and $d_1 =?$ (the value that your algorithm has to compute).

ii) The output of your problem should be the digit 0. Please note that 0 is the only value of d_1 between 0 and 10 for which $5 \times 1 + 4 \times 5 + 3 \times 2 + 2 \times 1 + 0 = 33$ is a multiple of 11.

GENERAL NOTES:
i) Attached to this assignment is a file called `ISBN.java`. Note that in this file there is a marked section where your code must go. The code outside of this area must not be modified.
ii) Note that this program is run by providing *input arguments*. If you are using Eclipse, please read this: `http://www.cs.colostate.edu/helpdocs/eclipseCommLineArgs.html`.

Question 2: What day of the week a date falls on? (25 points)

Do you know what day of the week you were born? It will be very cool if we can have a computer program that figures out that day for any given date. For example, have you ever thought what day of the week Isaac Newton was born? It may sound like a futile question, but you never know when you would need to impress someone with that kind of information. For this exercise, you will write a java program that takes three integers as input (representing the year y, month m and day d) and it will return a String representing the day of the week on which that date falls. You can fairly assume that the inputs represent a valid date. Regarding the months (m), please use 1 for January, 2 for February, 3 for March, and so on. To compute the day of the week, you must use the following formulas (based on a Gregorian calendar).

$y_0 = y - \frac{(14-m)}{12}$

$x = y_0 + \frac{y_0}{4} - \frac{y_0}{100} + \frac{y_0}{400}$

$m_0 = m + (12 \times \frac{14-m}{12}) - 2$

$d_0 = (d + x + \frac{31m_0}{12})\ mod\ 7$

Please note that the final result is stored in the variable d_0. A value of 0 denotes Sunday, 1 denotes Monday, 2 denotes Tuesday, and so forth.

Let's see an example to clarify what your program must output:

SAMPLE INPUT 1:
1953 8 2
SAMPLE OUTPUT 1:
SUNDAY

NOTES (for INPUT 1):
i) The input represents the date August 2, 1953. Please note that the parameters represent the year y, the month m and the day d, respectively.
ii) At the end of the execution, the values of the variables is as follows: $y_0 = 1953$, $x = 2426$, $m_0 = 6$ and $d_0 = 0$ (SUNDAY). We are giving to you these values such that you can debug your program in case it is not computing the right answer
iii) Since $d_0 = 0$, you must print the message "SUNDAY" in capital letters. You should not return any other String (it will make the automatization of the marking harder)

Lets see another example:
SAMPLE INPUT 2:
1893 12 26
SAMPLE OUTPUT 2:
TUESDAY

NOTES (for INPUT 2):
i) The input represents the date December 26, 1893.
ii) At the end of the execution, the values of the variables is as follows: $y_0 = 1893$, $x = 2352$, $m_0 = 10$ and $d_0 = 2$ (TUESDAY). We are giving to you these values such that you can debug your program in case it is not computing the right answer
iii) Since $d_0 = 2$, you must print the message "TUESDAY" in capital letters.

SAMPLE INPUT 3:

```
2000 1 1
```

SAMPLE OUTPUT 3:

```
SATURDAY
```

NOTES (for INPUT 3):
i) The input represents the date January 1, 2000.
ii) At the end of the execution, the values of the variables is as follows: $y_0 = 1999$, $x = 2483$, $m_0 = 11$ and $d_0 = 6$ (SATURDAY). We are giving to you these values such that you can debug your program in case it is not computing the right answer
iii) Since $d_0 = 6$, you must print the message “SATURDAY” in capital letters.

GENERAL NOTES:
i) Attached to this assignment is a file called `DateOnWeek.java`. Note that in this file, there is a marked section where your code must go.

Question 3: Judge Score (30 points)

Did you watch the Olympics in Rio de Janeiro? Did you notice that one common way to score the performance of an exercise routine was to average the scores of the judges after discarding the high and low scores? Here, we will create an algorithm that takes 4 integer values (representing judge scores) and prints their average, after discarding the highest and lowest score. For this exercise (and to be fair with students that do not have previous coding experience) you are not allowed to use the Math library, loops or arrays.

Let's see three examples of what your program should output:

SAMPLE INPUT 1:
```
1 2 3 4
```
SAMPLE OUTPUT 1:
```
2.5
```

NOTES (for INPUT 1):
i) The input represents the scores given by four judges.
ii) We should throw out the highest (i.e., 4) and the lowest (i.e., 1) score. Then, the remaining scores are 2 and 3. The average between those two scores is equal to 2.5 which should be the output of your program.

SAMPLE INPUT 2:
```
3 5 3 7
```
SAMPLE OUTPUT 2:
```
4
```

NOTES (for INPUT 2):
i) The input represents the scores given by four judges.
ii) We should throw out the highest (i.e., 7) and the lowest (i.e., one of the two 3's) score. Then, the remaining scores are 5 and 3. The average between those two scores is equal to 4 which should be the output of your program.

SAMPLE INPUT 3:
```
10 10 10 10
```
SAMPLE OUTPUT 3:
```
10
```

NOTES (for INPUT 3):
i) The input represents the scores given by four judges.
ii) We should throw out the highest (i.e., one of the four 10's) and the lowest (i.e., one of the four 10's) score. Then, the remaining scores are 10 and 10. The average between those two scores is equal to 10 which should be the output of your program.

GENERAL NOTES:
i) Attached to this assignment is a file called `JudgeScore.java`. Note that in this file there is a marked section where your code must go.

What To Submit

Please put all your files in a folder called Assignment1_ID, where 'ID' should be replaced by your McGill ID number. Zip the folder (please DO NOT rar it) and submit it in MyCourses. Inside your zipped folder there must be the following files (please do not submit any other files or folders, except from the ones listed below, especially .class files).

`DateOfTheWeek.java`
`JudgeScore.java`
`ISBN.java`
`Confession.txt` (optional) In this file, you can tell the TA about any issues you ran into doing this assignment. If you point out an error that you know occurs in your problem, it may lead the TA to give you more partial credit. On the other hand, it also may lead the TA to notice something that otherwise they would not.