

Assignment 2 : Checking SIN numbers

COMP-202, Summer 2016

Due: May 19th, 2016

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

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)

Write a program `Adder` that takes as input, using command line arguments (`args[0]` and `args[1]`) 2 numbers and adds them together. It should print the resulting value to the screen.

Warm-up Question 2 (0 points)

Write a program `MakingChange` that takes as input, using command line arguments (`args[0]`) 1 number. The program should print the correct change of that many cents. For example, if the number is 215, your program you print:

```
216 cents is
1 loonie
0 toonie
0 quarter
1 dime
1 nickel
0 penny
```

只做 Party
2 (下一
页)

Pretend it is still 2010 and there are pennies!

Hint: Use integer division to help figure out how many of each coin fit in.

Warm-up Question 3 (0 points)

Write a program `IndividualDigits` that takes as input, using command line arguments one number, which is assumed to be 8 digits. The program should print each digit, one at a time, on separate lines. For example, if the number is 12345678 your program should print.

```
1
2
3
4
5
6
7
8
```

Hint: To get the 1s digit of a number, you can take a number modulus 10.

Hint 2: To get the 10s digit of a number, you can first divide the number by 10 and then take the resulting value modulus 10. The idea is to “slide” the 10s digit into the correct position to apply the first hint. You can then do this trick on any column (e.g. hundreds, thousands, etc)

Warm-up Question 4 (0 points)

Write a program `DigitSum` that takes as input, using command line arguments one number, which is assumed to be 8 digits. The program should print the sum of all the digits. For example, if the input is `15610012` your program should print `16`

Part 2 (100 points): Understanding Social Insurance numbers

This question will be graded.

A social insurance number in Canada is a 9 digit number. It turns out that the last digit of the number is calculated based off of the first 8 numbers. This is designed as an easy way to detect (some) invalid social insurance numbers, before later on doing a more thorough check. The idea of this sort of calculation is called a *checksum*

In this assignment, you will generate the 9th digit from the first 8 digits. See instructions below:

Write a program called `SocialInsuranceChecksum` that takes as input, via `args[0]` a number which is assumed to be 8 digits long. (You do not need to add any code to check this.) The program should then calculate and print the 9th digit. You may also assume that all characters are between 1 and 9.

Your program *must* read everything from a single command line argument. (You must only use `args[0]`, not `args[1]`) You will need to extract the individual digits similar to the warm up question above.

A sample run in Dr Java is below:

```
run SocialInsuranceChecksum 08585827 <----The user typed this
6 <----The program printed this
```

Your program should not output anything other than the 1 digit. (As you are going, you will definitely want to temporarily add other print commands to see the values of variables. You should delete or comment these before handing in your assignment.)

Feel free to use your social insurance number (if you have one) to try it out. But you should not leave your SIN number in the version you hand it for your own security!

Generating the 9th digit

To generate the last digit, you will apply the following steps:

1. Create 8 variables and store into these 8 variables separately each of the 8 digits of the original number.
2. With counting from the left, take the 2nd, 4th, 6th, and 8th digits, multiply them each by 2 and add the result together.
3. Take the result from the previous step and add the 1st, 3rd, 5th, and 7th digits together.
4. Take this result modulus 10.
5. The final 9th digit should be such that adding it to the result at the previous step should equal 10.

For example: If the number is `08585827`, we start by taking the individual digits:

We multiply 8,8,8,7 by 2 and get 16,16,16,14, which added together gives us 62. We then add this to 0,5,5,2 which gives us 74. 74 modulus 10 is equal to 4. We choose the number 6 as the last digit since 4 plus 6 is 10.

What To Submit

You have to submit one zip file called `Assignment2_YourName.zip` with all your files in it to MyCourses under Assignment 1. If you do not know how to zip files, please ask any search engine or friends. Google might be your best friend with this, and a lot of different little problems as well.

`SocialInsuranceChecksum.java`

It is mandatory that you submit a file with the correct name and points will be deducted if you name your file anything other than `SocialInsuranceChecksum.java`. We are being strict to make things as easy as possible to mark so that TAs can focus on providing meaningful feedback.