Exercise 1:

Create an array with 1 million elements. Initialise each element to a random value in the range 1-10.

Write and execute a thread to print the sum of all the even numbers in the array.

Write and execute a thread to print the sum of all the odd numbers in the array.

Then print the contents of the array, replacing each maximal sequence of 1's with a single 1. For example, if the array contained the following elements {4, 5, 1, 1, 1, 5, 7, 7, 1, 4, 4, 1, 1, 8 etc} your code should print – 4, 5, 1, 5, 7, 7, 1, 4, 4, 1, 8 etc. Write and execute a thread to print the array (with the maximal sequence of 1's removed).

Exercise 2:

A palindrome is any number, word or phrase that reads the same backwards as it does forwards. For example:

- Navan
- Aviva
- Anna
- Noel sees Leon
- 78087

Take any positive integer of two digits or more, reverse the digits, and add to the original number. Now repeat the procedure with the sum until a palindromic number is encountered.

Using 56 as an example, you can see that it takes ***one*** iteration of the procedure to arrive at a palindrome.

56 + 65 = 121

Using 57 as an example, you can see that it takes ***two*** iterations to arrive at a palindrome.

57 + 75 = 132

132 + 231 = 363

Using 59 as an example, you can see that it takes ***three*** iterations to arrive at a palindrome.

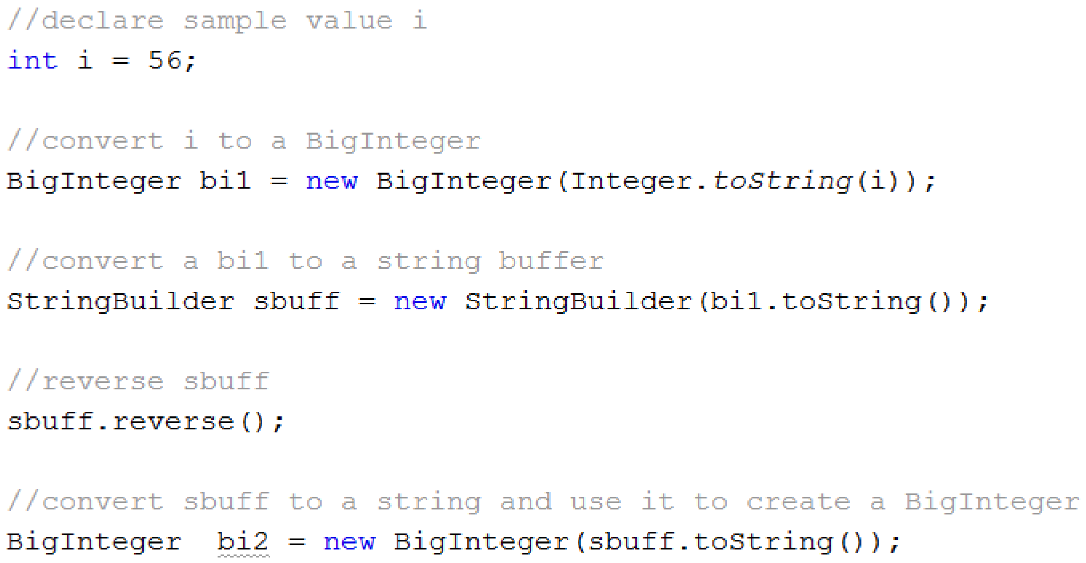
59 + 95 = 154

154 + 451 = 605

605 + 506 = 111

Write a program to output all the numbers below 10,000 that do not become palindromic after 50 iterations. You must also output how many numbers exist below 10,000 that do not become palindromic after 50 iterations (the answer is 249). This code must run as a thread within your application.

Code to consider:



```
//declare sample value i
int i = 56;

//convert i to a BigInteger
BigInteger bi1 = new BigInteger(Integer.toString(i));

//convert a bi1 to a string buffer
StringBuilder sbuff = new StringBuilder(bi1.toString());

//reverse sbuff
sbuff.reverse();

//convert sbuff to a string and use it to create a BigInteger
BigInteger  bi2 = new BigInteger(sbuff.toString());
```

Note:

The only websites you are permitted to visit during this assignment are:

1. Moodle.
2. Java API Documentation.

Remember all websites you visit within the college are logged and will be checked.

Fun Facts:

4994, which itself is a palindrome, does not (surprisingly) become palindromic after 50 iterations.

For numbers less than 10,000, 29.9% of all numbers turn palindromic in one iteration.

For numbers less than 10,000 over 80.5% turn palindromic in four or less iterations.

89 takes an unusually long time to become palindromic (24 iterations). It is the largest number of iterations of any number less than 10,000.

89 → 187 → 968 → 1837 → 9218 → 17347 → 91718 → 173437 → 907808 → 1716517 → 8872688 → 17735476 → 85189247 → 159487405 → 664272356 → 1317544822 → 3602001953 → 7193004016 → 13297007933 → 47267087164 → 93445163438 → 176881317877 → 955594506548 → 1801200002107 →8813200023188

Hence the need to use *BigInteger* in your solution.