CS1010J Programming Methodology
Tutorial 3: Repetition Statements

> People learn something every day, and a lot of times it's that what they learned the day before was wrong.
> *~ Bill Vaughan*

To students:

Two weeks of tutorials have passed and hence we have removed the usual preamble, since you should already know what is expected of you by now.

I. Manual tracing

1. Consider the following program.

```
import java.util.*;

class T3Q1 {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    System.out.print("Enter n: ");
    int n = sc.nextInt();

    int i = 1, count = 0;
    while (i < n) {
      if (n%i == 0) {
        count++;
      }
      i++;
    }

    System.out.println("count = " + count);
  }
}
```

(a) Assuming that the user enters 20, what is the final value of **`count`**?

(b) Assuming that the user always enters a positive number, describe the purpose of this program.

(c) If the final value of **`count`** is 1, what can be said about the input value **`n`**?

(d) Knowing what this program does, could you change the 'while' condition to make the program work a little more efficiently (i.e., take fewer iterations to compute the answer)?

2. Manually trace each of the following code fragments and write down its output.

(a)

```
int product = 0;
for (int i = 1; i < 10; i++) {
  product *= i;
}
System.out.println("product = " + product);
```

(b)

```
int sum = 0;
for (int i = 1; i < 1000; i *= 2) {
  sum += i;
}
System.out.println("sum = " + sum);
```

(c)

```
int count = 0;
for (int x = 1; x <= 6; x++) {
  for (int y = x + 1; y <= 6; y++) {
    count++;
  }
}
System.out.println("count = " + count);
```

II. Programming

3. **Conversion of loop constructs**

You have learned 3 loop constructs: **for**, **while** and **do-while**. For each of the following parts, a particular loop construct is used. Rewrite the loop construct using the other loop construct (without adding any **if-else** statement).

(a) Turn the following **do-while** loop into **for** loop.

```
int sum = 0, i = -5;

do {
  sum += i;
  i += 5;
} while (i < 100);

System.out.println("sum = " + sum);
```

(b) Turn the following **while** loop into **do-while** loop. Can you describe what the code fragment does?

```
Scanner sc = new Scanner(System.in);

System.out.print("Enter n: ");
int n = sc.nextInt();

while (n < 0) {
  System.out.print("Enter n: ");
  n = sc.nextInt();
}

System.out.println("n = " + n);
```

4. **[Problem Set 2 Exercise #02] Count Positive Integers**

Write a program **PS2_02_CountPositive.java** to read 5 integers from user input and count how many of them are positive.

```
Enter 5 integers: 2 5 7 0 3
Count = 4
```

```
Enter 5 integers: 0 -1 0 -3 -2
Count = 0
```

5. Programming with loops

(a) Write a method **`int sumUpMultiplesOf5(int n)`** to sum the first **`n`** positive multiples of 5. For example, if **`n`** is 4, then it computes 5 + 10 + 15 + 20 = 50.

```
Enter n: 4
Sum = 50
```

```
Enter n: 10
Sum = 275
```

Answer:

```
public static int sumUpMultiplesOf5(int n) {




}
```

(b) Write a method **`int sumUpNTerms(int n)`** to sum the first **`n`** terms of this series:

$$1 + 3 + 7 + 15 + 31 + \ldots (2^i - 1) + \ldots$$

For example, if **`n`** is 4 then it computes 1 + 3 + 7 + 15 = 26.

```
Enter n: 4
Sum = 26
```

```
Enter n: 10
Sum = 2036
```

Answer:

```
public static int sumUpNTerms(int n) {









}
```

6. **[Problem Set 2 Exercise #10] Collatz Problem**