Review Assignment 07
Submission Deadline 11:59 PM, Sunday, March 5th
Total 35 points

Instruction.

1. **Download the assignment sheet.**
2. **Enter your answer.**
3. **Upload your answer sheet.**

Question 1. Finding all occurrences of " " using a while-loop (12 points).

Let `wordInput` be a `String` variable. Suppose you want to construct a while-loop for identifying all the occurrences of " " in `wordInput`. We will use position as the search position, result for receiving the value from the indexOf() method, and count for counting the occurrences of " ". Fill in the blanks.

```
int result, position, count;
count = 0 ;    // initialization 2 points
position = 0; // initialization 2 points
do {
  result = wordInput.indexOf( " ", position );
  if (result >= 0) { // 3 points
    count ++;
    position = result + 1; // 2 points
} while (result >= 0); // 3 points
System.out.printf( "%d occurrences%n", count );
```

Question 2. Throwing three dices until all turn up 6Finding all occurrences of " " using a while-loop (9 points).

Suppose you want to run a simulation to figure out, when you roll three dice at a time, how many throws you must make to see 6 on all three. In the simulation, you mimic a roll of three dice drawing three random numbers; you repeat the process until all the three numbers generated are 6, and then you report the number of rolls. Fill in the blanks.

```
int rolls, n1, n2, n3;
rolls = 0;    // initialization 2 points
do {
  n1 = (int)(Math.random() * 6 + 1); // 3 points
  n2 = (int)(Math.random() * 6 + 1);
  n3 = (int)(Math.random() * 6 + 1);
  rolls ++;
} while ( !(n1 == 6 && n2 == 6 && n3 == 6) ); // 4 points
System.out.printf( "%d rolls%n", rolls );
```

Question 3. Contirnu4ed from Question 2 (3 points)
If you are to change the above code to one that uses a while-loop and with the same continuation condition, you need to assign an initial value to n1, n2, and n3. What initial would you give to them? You can assign the same value to all three. `(int)(Math.random() * 6 + 1);`

Question 4. The largest power of 2 not exceeding the input (11 points altogether).
Suppose you are to write a program that receives an integer from the user and compute the largest power of 2 that does not exceed the input. For example, the power in question is 2 if the input is either 2 or 3 and 3 if the input is one of 4, 5, 6, and 7. Suppose you are to write the code a shown below. Fill in the blanks so that the code works as desired.

```
int power, value;
Scanner in;
in = new Scanner( System.in );
System.out.print( "Enter value: " );
value = in.nextInt(); // 2 points
power = 0; // 2 points
while (Math.pow(2, power+1) <= value) { // 4 points
  power++;// 3 points
}
System.out.println( power );
```