Exercise 1. Guess the Output

Write the output of each of the following statements in the corresponding text boxes:

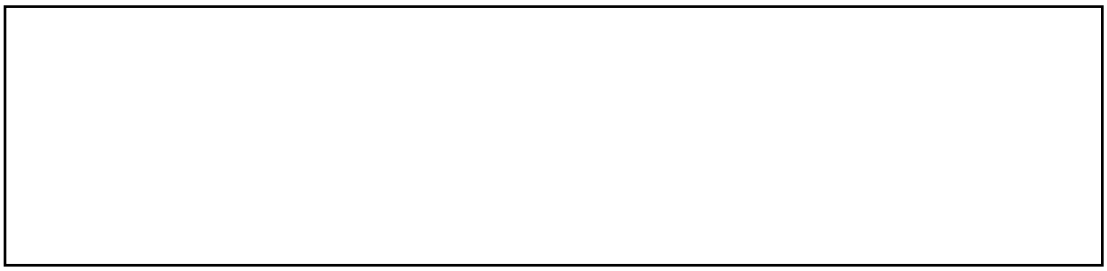
```
int number = 10;
while (number > 0) {
   System.out.println(number);
   number -= 1;
}
System.out.println(number);
```

```
int n = 100;
for (n = 0; n < 5; n++)
    System.out.println(n);
System.out.println(n);
```

```
String str = new String("Quiz 1 Soon");
int i = 0;
for (; i < str.length()-1; ++i) {
     System.out.println("At position " + ++i + " is " + str.charAt(i));
System.out.println(i++);}
```

```
int n = 10;
for (; n < 5; n++)
    System.out.println(n);
System.out.println(n);
```

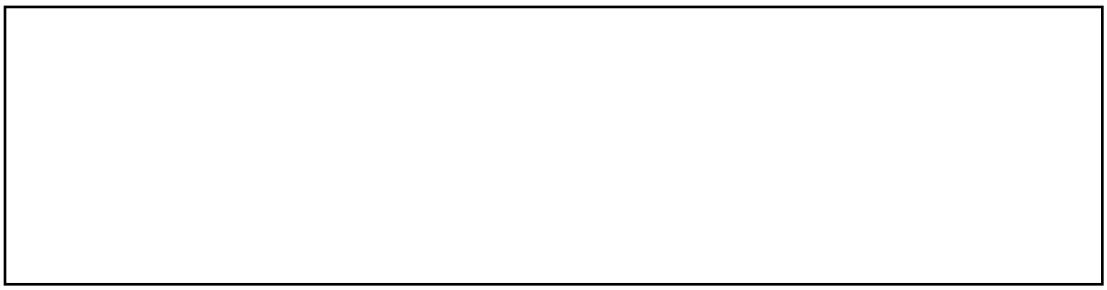
```
int i = 0, j, chars = 0;
String s = "Loops are fun";
while (i < 5) {
   j =0;
   while(j < s.length()) {
      j++;
      chars++;}
   System.out.println("Loop: " + i + " characters: " + chars);
   i++;}
```



Exercise 2. For to While

Write the equivalent code to the below nested for loops, using nested while loops:

```
for(int i = 0; i < 5; i++) {
      for(int j =0; j < 10; j++)
            System.out.print("-");
       System.out.println(); }
```



Exercise 3. Digits

Write a method named **allEvenDigits** that returns whether every digit of a positive integer is even. Your method should return **true** if the number consists e**ntirely of even digits** and false if any of its digits are odd. **Note that, you are not allowed to change the integer into a string.**

For example, **allEvenDigits** (20426) returns **true** but **allEvenDigits**(14654) returns **false**.

Hint: You can pull apart a number into its digits using / 10 and % 10.

Exercise 4. Rock-Paper-Scissor

Design and implement a JAVA application that plays the Rock-Paper-Scissors game against the computer. When played between two people, each person picks one of three options (usually shown by a hand gesture) at the same time, and a winner is determined. In the game, Rock beats Scissors, Scissors beats Paper and Paper beats Rock. The program should randomly choose one of the three options (without revealing it), then prompt for the user's selection. At that point, the program reveals both choices and prints a statement indicating if the user won, the computer won or if it was a tie. Continue playing until the user chooses to stop, then print the number of user wins, losses and ties. A sample run:

```
Rock, Paper, or Scissors? Paper
Player: Paper
Computer: Paper
Winner: It's a tie!
Again? (yes/no): yes
Rock, Paper, or Scissors? Rock
Player: Rock
Computer: Scissors
Winner: Player
Again? (yes/no): yes
Rock, Paper, or Scissors? Scissors
Player: Scissors
Computer: Rock
Winner: Computer
Again? (yes/no): no
******************
Wins: 1
Losses: 1
Ties: 2
******************
```

Exercise 5. Undouble

Write a static method called **undouble** that takes a string as a parameter and that returns a new string obtained by replacing every pair of repeated adjacent letters with one of that letter. For example, the String "bookkeeper" has three repeated adjacent letters ("oo", "kk", and "ee"), so `undouble("bookkeeper")` should return the string "bokeper". Your method should be case insensitive and you may assume that no letter appears more than two times consecutively.

Below are more sample calls:

```
undouble(“odegaard”) → odegard

undouble(“oops”) → ops

undouble(“Mississippi”) → misisipi
```

Exercise 6. Factors of 2

Write a method named showFactors2 that shows the factors of 2 in a given integer. For example:

```
showFactors2(7)   →  7 = 7
showFactors2(18)  → 18 = 2 * 9
showFactors2(68)  → 68 = 2 * 2 * 17
showFactors2(120) → 120 = 2 * 2 * 2 * 15
```

Exercise 7. Dices Sum

Write a method that prompts the user for a desired sum, then repeatedly rolls two six-sided dice until their sum is the desired sum.

A sample run:

```
Enter desired dice sum: 9
5 and 1 = 6
3 and 3 = 6
2 and 5 = 7
6 and 5 = 11
6 and 3 = 9
```