# Loops and Conditionals Challenge Problems

1. Given 2 ints, a and b, print true if one of them is 10 or if their sum is 10.

2. Given an int n, return true if it is within 10 of a multiple of 100. Note: Math.abs(num) computes the absolute value of a number.

3. Given a string str, if the string starts with "f" return "Fizz". If the string ends with "b" return "Buzz". If both the "f" and "b" conditions are true, return "FizzBuzz". In all other cases, return the string unchanged. Hint: Use substring method. Given a string str, a substring of str of the first 5 letters of str would be str.substring(0, 5)

4. We'll say a number is special if it is a multiple of 11 or if it is one more than a multiple of 11. Return true if the given non-negative number is special.

5. The squirrels in Palo Alto spend most of the day playing. In particular, they play if the temperature is between 60 and 90 (inclusive). Unless it is summer, then the upper limit is 100 instead of 90. Given an int temperature and a boolean isSummer, return true if the squirrels play and false otherwise.(units do not matter). The temperature is a value given by user.

6. Given 3 int values, a b c, return their sum. However, if one of the values is the same as another of the values, it does not count towards the sum.

7.Tell the user to define a range and print the sum of all numbers in that range inclusively. For example, if the user inputs 1 and 5 the output would be 15 because 1 + 2 + 3 + 4 + 5 = 15. Is there a way to do this without loops?
**Something to keep in mind**: The first number input by the user does not necessarily have to be lower than the second.

8. Print the factorial of a number input by the user. **Hint**: Factorial of 5 = 5! = 5 * 4 * 3 * 2 * 1.

9. Two numbers are entered using the scanner class. Write a program to find the power of one number raised to another.. Do not use Java's inbuilt power class to do this.*Extra: Create one option where if the user types a certain response, the first number is raised to the second number and another option where the second number is raised to the first number

10. Compute the natural logarithm of 2, by adding up to n terms in the series
1 - 1/2 + 1/3 - 1/4 + 1/5 -... 1/n
where n is a positive integer and input by user.(Hint:set up a variable to account for the sign that is initialized to 1.