1. Create a variable to hold the number (integer) 31. Now print a statement that says: "**August has 31 days.**" using the variable concatenated INTO the print statement.
2. Create variables to hold the strings: "September", "April", "June" and "November. Now create a variable called statement that will hold a strings that says: "**Thirty days have September, April, June, and November.**". Be sure to concatenate the variables in for the month names. Print the statement variable.
3. Create a variable to hold the string "October". Now write an if/else statement that tests if that variable says "October". If it does, print "**31 days**". If it does not, print "**30 days**"
4. Create a variable to hold the number 70. Now create two if statements:  
   1st: if the number is divisible by 5 and by 10, then print "Number is divisible by 5 and 10."  
   2nd: if the number is divisible by 7 or by 2, then print "Number is divisible by 7 or by 2"
5. Create a FOR loop that prints the numbers 1 through 10.
6. Create a FOR loop that prints the numbers 6 through 20, but only prints every THIRD number.
7. You will be creating a nested loop like we did in class but with a twist. Ultimately, you want 5 sets to print, each one with the output "**Set : (set number)**" each time a new set is looped into. Then before you print all the numbers in that set, you want to test if the number is divisible by 2. If it is, print a string 5 times that looks like this: "**Student (number)**".  So it would have 5 sets... some of them would have strings printed in them, some would not. The ones that would have strings in them would show output like:  
   "  
   **Set: 1  
   Set: 2  
   Student 1  
   Student 2  
   Student 3  
   Student 4  
   Student 5  
   Set: 3**"  
   ...
8. Use a while loop to print all numbers from 1 to 10 which are divisible by 3. Use a while loop and if statement.
9. Create a function called doCalc and it will receive 3 parameters: num1, num2, and num3. Within the function, add num1 and num2 and then multiply them by num3.  
   Print the answer from within the function. Call doCalc once and pass in the arguments: 10, 12, 40.
10. Create a function called creditsPerQuarter. It will receive parameters: student, numCreds, numCourses. You will pass in a students name, the number of credits each course they are taking is and the number of courses they are taking. Within the function, you will calculate the total number of credits that student has that quarter. You will RETURN the following string: (student name) is taking (number of courses) courses and each course is (number of credits each course is) for a total of (calculated total credits). For example: If the student is Bob and he is taking 4 courses with 3 credit each it would read exactly:  
    "Bob is taking 4 courses and each course is 3 credits for a total of 12."

         You will call creditsPerQuarter three times and pass in these arguments:

* Student Name: Wade, Credit per Course: 3, Number of courses: 4
* Student Name: Jane, Credit per Course: 3, Number of courses: 5
* Student Name: Evan, Credit per Course: 5, Number of courses: 3

         Your output should look like this exactly:

       Wade is taking 4 courses and each course is 3 credits for a total of 12.  
       Jane is taking 5 courses and each course is 3 credits for a total of 15.  
       Evan is taking 3 courses and each course is 5 credits for a total of 15.