**Practice Questions – May 6, 2024**

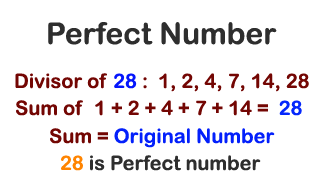
Try to solve each of the following using JavaScript. If you need to print something, use console.log(), which will print the argument(s) you give it.

1. Create a variable label and assign it the value "keyincollege". Create another variable tld and assign it "ca". Create a third variable domainName that combines label and tld to produce the value "keyincollege.ca".
2. Create a variable isKeyin and assign it a boolean value (true or false) depending on whether or not domainName is equal to "keyincollege.ca". HINT: use === and don’t write true or false directly.
3. Create a variable isNotKeyin and assign it the inverse boolean value of isKeyin. HINT: if isKeyin is true, isNotKeyin should be false.
4. Create four variables byte1, byte2, byte3, byte4, and assign each of these a value in the range 0-255.[You can assign any random numbers to each byteN variable, like 198 or 110 or 1 or any number in the range of 0-255.
5. Create a variable ipAddress and assign it the value of combining your four byteN variables together, separated by ".". For example: "192.168.2.1".
6. Write code to display the multiplication table of a given integer.    
   Define any variable and assign it a value, like in following example: your number is 15. I will test your code with changing the value of the variable…

var table = 15;// you need to define a variable like this and display its table…

*Expected Output*:  
15 X 1 = 15  
...  
...  
15 X 10 = 150

1. Write code to display all even and odd numbers from first 100 numbers by using for loop and if condition.
2. By using the same concept of above question no.7, display the sum of all even numbers from 100 numbers.
3. A perfect number is a number whose divisors sum is equal to number itself:



Write code to check whether a given number is “perfect number” or not. You can again just define a variable and apply logic to check it.

1. Write a program to determine whether a given number is prime or not

Chart

Description automatically generated