Java AP - Bitwise Operators
Assignment #22

This is an assignment to help you practice with numbering systems and bitwise operators. There are four functions for you to write to complete the assignment.

Continuing from the numbering system chapter we will utilize bitwise operators to convert between representations of integers in various base systems.

Function	`char[] DecToBinary(int number)`
Description	Given a 32 bit integer representing a decimal value, this function will return a character array containing the binary representation of that decimal value. **Example:** Given a value 10 which is [...0000000001010] in binary, ***DecToBinary(10)*** returns an array containing 32 characters: **{... '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '1', '0' }**

Hint:
In order to check if the 1st bit is set or not:

number	0	0	0	0	1	0	1	0
&								
mask	0	0	0	0	0	0	0	1
result	0	0	0	0	0	0	0	0

If the **result** is equal to **zero** then the specific bit is **not set (0)**, otherwise the bit is **set (1)**.

In order to check if the 2nd bit is set or not:

number	0	0	0	0	1	0	1	0
&								
mask	0	0	0	0	0	0	1	0
result	0	0	0	0	0	0	1	0

Because the **result is not zero**, we know that the **2nd bit is set.**

Note: In order to check all the bits, you will keep shifting the 1 value in the mask one position to the left and check again.

Function	`int BinaryToDec(char number[])`
Description	Given a character array containing 32 characters representing a value in binary, this function will return an integer representing the binary value in decimal. **Example:** Given the following **array** of 32 characters [...0000000001010] (where **"..."** is all zero values) which represents the binary value, ***BinaryToDec(array)*** returns an integer containing the value **10.**

Hint:
In order to set a bit in an integer you do the following: ***number = number ! mask***
In order to set the 1st bit to 1:

number	0	0	0	0	0	0	0	0
|								
mask	0	0	0	0	0	0	0	1
result	0	0	0	0	0	0	0	1

In order to set the 2nd bit to 1:

number	0	0	0	0	0	0	0	1
|								
mask	0	0	0	0	0	0	1	0
result	0	0	0	0	0	0	1	1

NOTE: You may NOT use division or modulus (%,/) operators for this assignment. This assignment is specifically targeting bitwise operators.

An ***output.txt*** file is given to you so that you can compare your output. **(I expect the exact output, so use a diff tool to check your output with mine).**

What to submit
You must submit the Java file (**Conversion.java)** in a single .zip file named **"Assignment22.zip"** (go to the class page on moodle and you will find the assignment submit link). **Do not submit any other files than the ones listed.**

Special note:
The due date/time posted is the positively latest you are allowed to submit your code. Since the assignment can easily be completed well before the deadline, you should strive to turn it in as early as possible. If you wait until the deadline, and you encounter unforeseen circumstances (like being sick, or your car breaking down, or something else), you may not have any way to submit the assignment on time. **Moral: Don't wait until the last day to do your homework.**

Extra Challenge (which you will **NOT** submit):

If you would like to challenge yourself a little bit more, write the below two functions.

Function	`char[] DecToHex(int number)`
Description	Given a 32 bit integer representing a decimal value, this function will return a character array containing the hexadecimal representation of that decimal value. **Example:** Given a value **1000** which is **[3E8]** in hexadecimal, ***DecToHex(1000)*** returns an array containing 8 characters: **{'0', '0', '0', '0', '0', '3', 'E', '8' }**

Function	`int HexToDec(char number[])`
Description	Given a character array containing 8 characters representing a value in hexadecimal, this function will return an integer representing the hexadecimal value in decimal. **Example:** Given the following **array** of 8 characters **[000003E8]** which represents the hexadecimal value, ***HexToDec(array)*** returns an integer containing the value **1000.**

An ***output-challenge.txt*** file is given to you so that you can compare your output. Check the **"Conversion Challenge"** folder.