Computer Science 1-Honors	Lab 14C 1-Day Minor Python Assignment
Number Systems – Part II	80, 100 & 110 Point Versions

Assignment Purpose:

The purpose of this lab assignment is to gain more understanding of string processing and "Number Systems"; specifically, by converting binary numbers to decimal.

In this assignment, you will convert provided *binary* numbers into *decimal*. Even though the binary numbers are provided, you need to write your program in such a way that it would work for any numbers, instead of just the ones that are provided.

Lab 14C Student Version	Do not copy this file, which is provided.

```
 1 # Lab14Cst.py
 2 # "Number Systems - Part II"
 3 # This is the student, starting version of Lab 14C.
 4
 5
 6 def heading():
 7     print()
 8     print("*********************************************")
 9     print("Lab 14C, Number Systems - Part II")
10     print("80 Point Version")
11     print("By: JOHN SMITH") # Substitute your own name here.
12     print("*********************************************")
13     print("\n")
14
15
16 def bin2dec(bin):
17     # precondition:  bin is a string storing an 8-bit sequence of 1s & 0s.
18     # postcondition: The decimal equivalent of bin is returned.
19     dec = 0
20     if bin[0] == '1':
21         dec += 128
22
23
24
25
26     return dec
27
28
29
```

```
30 ##########
31 #  MAIN  #
32 ##########
33
34 heading()
35 print(bin2dec("11001000"))
36 print(bin2dec("01100100"))
37 print(bin2dec("00110010"))
38 print(bin2dec("00011001"))
39
40 # Required for the 110 Point Version Only:
41 #print(bin2dec("1000000001"))
42 #print(bin2dec("1101"))
43 #print(bin2dec("101"))
44 #print(bin2dec("11"))
45 #print(bin2dec("0"))
46 #print(bin2dec("1100100000000"))
47
```

Specifics and Sample Output for the Student Provided File

You are provided with a complete **heading** procedure and a partial **bin2dec** function. This is the function that is supposed to convert a *binary* number to *decimal*. Remember, "decimal" is also called "base-10". This is the number system we use which consists of 10 digits (0,1,2,3,4,5,6,7,8,9). Binary is also called "base-2". This is the number system used by computers and consists of 2 digits (1 & 0). Right now, the **dec2bin** function is not complete, so it just returns **128** for 8-bit binary numbers that start with **1** and **0** otherwise.

```
 ----jGRASP exec: python Lab14Cst.py

*******************************************
Lab 14C, Number Systems - Part II
80 Point Version
By: JOHN SMITH
*******************************************


128
0
0
0

 ----jGRASP: operation complete.
```

NOTE: The **bin2dec** function that you are writing in this lab is the inverse of the **dec2bin** function that you wrote in Lab 12B.

80 and 100 Point Versions Specifics and Output

Both the 80 and 100 Point Versions require that the **dec2bin** function works for any 8-bit number. For the 80 Point Version, you are allowed to use 8 separate **if** statements, one for each bit. For the 100 Point Version, you are required to use a loop to make the function more efficient.

80 and 100 Point Versions Output

Both the 80 and 100 Point Versions have the same output.

```
 ----jGRASP exec: python Lab14Cv80.py

****************************************
Lab 14C, Number Systems - Part II
80 Point Version
By: JOHN SMITH
****************************************


200
100
50
25

 ----jGRASP: operation complete.
```

110 Point Version Specifics

The 110 Point Version requires everything from the 100 Point Version, except now the **dec2bin** function needs to work for any size binary number, not just the ones that have a fixed size of 8 bits. To properly test the 110 Point Version, you will need to uncomment the 6 **print** commands at the end of the program.

110 Point Version Output

```
 ----jGRASP exec: python Lab14Cv110.py

*********************************************
Lab 14C, Number Systems - Part II
110 Point Version
By: JOHN SMITH
*********************************************


200
100
50
25
513
13
5
3
0
6400

 ----jGRASP: operation complete.
```