Review Exercise: Encoding

|  |  |
| --- | --- |
| Binary | Hexadecimal |
| 0000 |  |
| 0001 |  |
| 0010 |  |
| 0011 |  |
| 0100 |  |
| 0101 |  |
| 0110 |  |
| 0111 |  |
| 1000 |  |
| 1001 |  |
| 1010 |  |
| 1011 |  |
| 1100 |  |
| 1101 |  |
| 1110 |  |
| 1111 |  |

Use **capital** (not lower-case) letters for hexadecimal digits above nine. Show all work.

1. ***Complete*** the Hexadecimal column in the table on the right.
2. ***Convert*** the following bits into hexadecimal:

11001111 10100111 00100011

* 1. ***Convert*** 19410 to binary. Convert back to decimal to check your answer.
  2. ***Convert*** your answer from the previous part to hexadecimal.

1. a. ***Find*** the number of values a 6-bit binary number can hold.  
     
   1. ***Find*** themaximum value of a 6-bit binary number.
2. ***Approximate*** how many values a 40-bit number can hold.
3. ***Approximate*** how many bits are needed to store numbers ranging from 1 through 1015.

1. Consider this message in hexadecimal shorthand:

0c 37 a2 4c fb f5 5b 6e ad 72 42 ce b4 e6 25 4f a0 d9 76 d0 07 9f 8d …

* 1. ***Circle*** the first two bytes in the message above
  2. ***Write*** the value of these bytes in **binary**.
  3. ***Write*** the value of these bytes in **decimal**.

1. ***Convert*** the following message from ASCII codes to ASCII characters

44 41 44 0D 0A 42 45 41 44 0D 0A

1. ***Write*** the hexadecimal shorthand for how the bytes object is stored in Python

b'Ebb\r\n'