Python Number Converter

Generally speaking, practiced skill cannot be easily forgotten. It is far better to go through the process and practice converting a number, rather than to memorize the process.

Before we get started, have a look at the Tools Page to get started with a Python Interpreter we could use for this exercise.

Number to Printable Hex

```
def nibble_to_ascii(nibble: int) -> str:
  """
  This is a comment
  Input: Nibble (4-bits)
  Output: Single character HEX as a string
  Example: Input = 10, Output = 'A'
  Example: Input = 8,  Output = '8'
  """
  table = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F']
  return table[nibble]
def to_hex(number: int) -> str:
    """
    This is a comment
    Input: Number (integer)
    Output: String
    Example: Input = 43605, Output = "0xAA55"
    """
    answer = ""

    # Forever loop
    while True:
        # Integer divide using the // operator
```

```
        quotient = number // 16
        # Get the remainder using the % operator
        remainder = number % 16

        # Accumulate result
        answer = nibble_to_ascii(remainder) + answer
        # Set the number we need to use for next time
        number = quotient

        # We break the "loop" when division turns to zero
        if (quotient == 0):
            break

    return "0x" + answer
print(to_hex(123456789))
print(to_hex(0b1010101))print(to_hex(0xDEADBEEF))
```

Exercise

Write a function `to_binary()` that takes a number, and returns the string equivalent version of the number in binary. You can borrow the template above of `to_hex()` function and most of the logic might be similar except that we would be dividing number by 2 rather than 16.

Revision #5
Created 1 year ago by Preet Kang
Updated 1 year ago by Preet Kang