

Homework 3

CS 2000: Python Programming Language

Wassnaa AL-mawee

Western Michigan University

February 5, 2018

In this assignment we will use **user_defined_functions** to manipulate bits values for comparisons and calculations. It is worth 5 points and is due (02/16/2018).

1. The program will first prompt the user for an operation. The possible operations are |, &, and ^ (**or**, **and**, **xor**). If q is entered as the operation, then the program should quit. If any other operation is input, the program should print the message "Please enter |, &, ^, or q". The program should then output the Enter operation prompt again on the next line. After the operation has been entered, the user will be prompted for a number of integers. The user must enter a valid integer **greater than one**. Then prompt the user for that many integers in the manner shown below. The integers must be formatted in hexadecimal, and must have a maximum of 8 hexadecimal digits. Input integers with less than 8 digits should be assumed to have leading zeros. Any invalid input should result in the message "Please enter an 8-digit hexadecimal integer", followed by the prompt for that integer on the next line. After receiving valid input, the program should then perform the operation on the numbers in the form *firstnum operator secondnum [operator thirdnum etc]*. The program should then output the entire operation (including the operands, operation, and result) in both binary and hexadecimal. The hexadecimal values should be padded so that they have 8 digits. The binary values should be

padded so that there are 32 total digits, grouped into clusters of 8 digits. Below is the format for a sample operation, where red text indicates user-entered values.

2. Example one:

Enter operation: |
Enter number of integers: 2
Enter integer 1: 000000FF
Enter integer 2: 30C01101

Hexadecimal operation:

000000FF
| 30C01101
= 30C011FF

Binary operation:

00000000 00000000 00000000 11111111
| 00110000 11000000 00010001 00000001
= 00110000 11000000 00010001 11111111

Enter operation: q

3. Example two:

Enter operation: @
Please enter |, &, ^, or q
Enter operation: &
Enter number of integers: 3
Enter integer 1: DF5383
Enter integer 2: Z0238984
Please enter an 8-digit hexadecimal integer
Enter integer 2: 908070605040

Please enter an 8-digit hexadecimal integer

Enter integer 2: **C5204**

Enter integer 3: **F13FB**

Hexadecimal operation:

00DF5383

& 000C5204

& 000F13FB

= 000C1200

Binary operation:

00000000 11011111 01010011 10000011

& 00000000 00001100 01010010 00000100

& 00000000 00001111 00010011 11111011

= 00000000 00001100 00010010 00000000

Enter operation: **q**

4. For the purpose of this assignment, you can use **only** three Python built-in functions: *input()*, *int()*, *hex()*

5. You need to define **three** User_defined_functions as follow:

- **check_hex():**
This function validates the input integer (hexadecimal format). It return a value.
- **hex_operation():**
This function fixes the length of hexadecimal number, performs the operations on the given numbers, and prints out the hexadecimal operation. It returns two values (list of fixed length hexadecimal numbers, operation) to the main program.
- **hex_binary():**
This function takes the returned values from

(hex_operation()). It converts the hexadecimal numbers into binary representation (The binary values should be padded so that there are 32 total digits, grouped into clusters of 8 digits (**Do not use bin()**). Also, it performs the operations on the binary numbers, and prints out the binary operation. It returns no value.

6. A major **goal** of this assignment is to test your **attention** to detail. Therefore, you must follow the output format shown above exactly. Labels in your program must exactly match the labels shown above. The same number of spaces that are used above must be used in your program. In particular, there should be one space after the labels before the user input. The operation results should be indented by two spaces, and there should be two spaces between each group of 8 binary digits. There will be no spaces on the right-hand side of any line after the output.
7. The program will be graded as follow:
 - Program correctness (3 pts)
 - Quality of design/readability/ Output formatting (2 pts)
8. What to turn in:

(Remember READABILITY COUNTS!)

- You will be turning in a print out of the following information :
 - The two provided examples (**Hint: I will test your code on different examples for program correctness ☺**).
 - Attach to your output a printed copy of the code you've written.
- On e-Learning, submit your Python file whose name

<hw#3_LastName.py>. Please format your Python code in the following manner:

```
# Name: <your name here>
# Date: <#/#/#>
# Homework: <#>
# Your code
```