Eric MAMPHEY

1610063

2.14: Stretch Tasks

1. What are binary numbers?

Ans: It is a [number](https://en.wikipedia.org/wiki/Number) expressed in the **base-2 numeral system** or **binary numeral system**, which uses only two symbols: typically "0" ([zero](https://en.wikipedia.org/wiki/Zero)) and "1" ([one](https://en.wikipedia.org/wiki/One)).

1. Without using Python or an online calculator, convert the number 42 into binary. Make sure that you show all your calculations.

Ans: 42/2 = 21 r 0

21/2 = 10 r 1

10/2 = 5 r 0

5/2 = 2 r 1

2/2 = 1 r 0

1/2 = 0 r 1

Hence 101010

1. Enter the following code in your Python file

print(bin(42))

Ans: 0b101010

1. What do you think the 0b at the beginning of the output means?

Ans: a prefix indicating binary format, common in programming languages

1. Without using Python or an online calculator, convert 1011011 into an integer.

Ans: 1011011 = 1x26 + 0x25 + 1x24 + 1x23 + 0x22 + 1x21 + 1x20

= 1x64 + 0x32 + 1x16 + 1x8 + 0x4 + 1x2 + 1x1

= 64 + 0 + 16 + 8 + 0 + 2 + 1

= 91

1. Enter the following code into your Python file

print(int(0b1011011))

You will notice that we have also put the 0b in the brackets, to identify to the interpreter that the number is binary. Try entering the code without the 0b and see what happens.

Ans: 1011011, It was treated as an integer by the IDE

1. What are octal numbers?

Ans: Octal Number System uses eight digits, 0,1,2,3,4,5,6,7.

Also called base 8 number system. Each position in an octal number represents a 0 power of the base (8). Example: 8 0 4 Last position in an octal number represents an x power of the base (8). Example: 8 x where x represents the last position - 1

1. Without using Python or an online calculator, convert the number 42 into octal. Make sure that you show all your calculations.

Ans: 42/8 = 5.25 (0.25 x 8 = 2)

5/8 = 0.625 (0.625 x 8 = 5)

oct42 = 52

1. Python has an in-built function which will convert numbers into octals. Establish what this command is, then use it to verify you have calculated the correct answer to question 8

Ans: print(oct(42))

0o52

1. The number 174 has been written in octal. Without using Python or an online calculator, convert 174 into an integer.

Ans: (4 x 1) + (7 x 8) + (1 x 64)

4 + 56 + 64

124

1. Using the int() command, check that your calculation for converting the octal 174 into an integer

Ans: print(int(0o174))

124

1. What are hexadecimal numbers?

Ans: hexadecimal (also base 16 or hex) numeral system is a [positional numeral system](https://en.wikipedia.org/wiki/Numeral_system#Positional_systems_in_detail) that represents numbers using a [radix](https://en.wikipedia.org/wiki/Radix) (base) of 16. Unlike the [common way](https://en.wikipedia.org/wiki/Decimal) of representing numbers using 10 symbols, hexadecimal uses 16 distinct symbols, most often the symbols "0"–"9" to represent values 0 to 9, and "A"–"F" (or alternatively "a"–"f") to represent values 10 to 15.

1. Without using Python or an online calculator, convert the number 42 into hexadecimal. Make sure that you show all your calculations.

Ans: 42/16 = 2, remainder is 10

2/16 = 0, remainder is 2

= 2A

1. Python has an in-built function which will convert numbers into hexadecimals. Establish what this command is, then use it to verify you have calculated the correct answer to question 13

Ans: print(hex(42))

0x2a

1. The number 2F67 has been written in hexadecimal. Without using Python or an online calculator, convert 2F67 into an integer.

Ans: (2F67)16 = 2x163 + Fx162 + 6x161 + 7x160

= 2x4096 + 15x256 + 6x16 + 7x1

= (12135)10

1. Using the int() command, check that your calculation for converting the hexadecimal into an integer

Ans: print(int(0x2F67))

12135

1. Enter the following code into your Python file:

print(1.2-1.0)

What do you think the result should be? Is that the answer you get? If not, why?

Ans: 0.19999999999999996

For the following questions, you will be using two variables, string1 and string2. You will need to define them at the beginning of your file. Make string1 equal to “hello” and string2 equal to “world”. For each of these questions, add the code to your file, and run it. Before you move on, make sure that you understand what each line of code is doing.

1. Add the following code to your python file

qn18\_string = string1 + string2

print(qn18\_string)

Ans: cannot seem to figure it out. Below is what I did and because it seems not to be correct, everything else after that would not work.

def string1():  
 print("hello")  
def string2():  
 print("world")  
string1()  
string2()  
  
qstn\_18 = string2() + string1()  
qstn\_18  
  
qn21\_string = len(string1())  
print(qn21\_string)

1. Adapt question 1 so that there is a space between the two words
2. Add the following code to your python file

qn20\_string = string1 + "\n" + string2

print(qn20\_string)

1. Add the following code to your python file

qn21\_string = len(string1)

print(qn21\_string)

1. Add the following code to your python file

print(string1 \* 5)

1. Adapt the print line for the question 5, so that each word has a space afterwards.
2. Add the following code to your python file

print(string1.upper())

1. Add the following code to your python file

qn25\_string = "HELLO WORLD"

print(qn25\_string.lower())

1. Add the following code to your python file

qn26\_string = "This is a sentence"

words = qn26\_string.split()

print(words[0])

print(words[1])

print(words[2])

print(words[3])

1. Add the following code to your python file

qn27\_string = "This-is-a-delimited-sentence"

words = qn27\_string.split("-")

print(words[0])

print(words[1])

print(words[2])

print(words[3])

print(words[4])

1. Enter the following code in your Python file. The id() command allows us to print the location the variable is stored within memory.

string3 = "a string"

string4 = "a string"

print("string3 id: "+str(id(string3)))

print("string4 id: "+str(id(string4)))

string3 += " updated"

print(string3)

print("string3 id: "+str(id(string3)))

print("string4 id: "+str(id(string4)))

What do you think the above code shows?

1. What does immutability mean in relation to Python?

1. Enter the following code in your Python file:

print("tim == tie: " + str("tim" == "tie"))

print("tim == tim: " + str("tim" == "tim"))

print("free != freedom: " + str("free" != "freedom"))

print("arrow > aron: " + str("arrow" > "aron"))

print("right >= left: " + str("right" >= "left"))

print("teeth < tee: " + str("teeth" < "tee"))

print("yellow <= fellow: " + str("yellow" <= "fellow"))

print("abc > '': " + str("abc" > ""))

What do each of these print statements tell you about how Python compares strings?