

Week 3 Worksheet Solutions

- (1) Suppose `todo` is an existing variable whose value is a list. Write code that will print "task overload!" if `todo` has 10 or more elements, and otherwise will print "Begin working on" followed by the first element of the list.

*Answer:*

```
if (len(todo) >= 10):
    print("task overload!")
else:
    print("Begin working on", todo[0])
```

- (2) Suppose the variable `c` has value that is a string of length 1 (i.e. a character). Write code to determine whether `c` is an alphabet character (A-Z,a-z), a digit (0-9), or other type of character. It should print a message stating the category that `c` belongs to. HINT: A solution using only the concepts we've discussed so far, but not involving too much typing, might begin

```
alphabet_chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
```

*Answer:*

```
alphabet_chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
numeric_chars = "0123456789"

if c in alphabet_chars:
    print("Alphabet Character")
elif c in numeric_chars:
    print("Numeric Character")
else:
    print("Other character type")
```

- (3) Using only the string variables `s` with value "knights" and `t` with value "rook", write expressions that produce the given string. In your expressions, string literals are forbidden!

- (a) "night"

*Answer:*  
`s[1:-1]`

- (b) "took"

*Answer:*  
`s[-2] + t[1:]`

- (c) "rig"

*Answer:*  
`t[0] + s[2:4]`

- (d) "ghost"

*Answer:*  
`s[3:5] + t[1] + s[-1] + s[-2]`

(4) Write a script that reads a string from the keyboard that consists of a float literal followed by either the character "C" or "F", for example 25.2C or 87.5F. The program should store the float in a variable `temp` and the final character in a variable `scale`, and then print a message depending on the input, as follows:

- If `scale` is "F" and `temp` is greater than or equal to 90, print "It is hot"
- If `scale` is "F" and `temp` is less than 90 but greater than or equal to 40, print "It is nice"
- If `scale` is "F" and `temp` is less than 40, print "It is cold"
- If `scale` is "C" and `temp` is greater than or equal to 32.2, print "It is hot, but at least you're using the metric system"
- If `scale` is "C" and `temp` is less than 32.2 but greater than or equal to 4.4, print "It is nice, like the metric system"
- If `scale` is "C" and `temp` is less than 4.4, print "It is cold"

Hint: An efficient way to write this is to put one `if` inside of another.

*Answer:*

```
tempstr = input("Enter a temperature followed by scale (C or F): ")
temp = float(tempstr[:-1])
scale = tempstr[-1]

if scale == "F":
    if temp >= 90:
        print("It is hot")
    elif temp < 90 and temp >= 40:
        print("It is nice")
    else:
        print("It is cold")
elif scale == "C":
    if temp >= 32.2:
        print("It is hot, but at least you're using the metric system")
    elif temp < 32.2 and temp >= 4.4:
        print("It is nice, like the metric system")
    else:
        print("It is cold")
```

(5) Write a script similar to the previous one, but which simply reads a temperature in either Celsius or Fahrenheit and then prints it in Celsius. For example, if the input is 32F it should print 0.0C, and if the input is 41.5C it should print 41.5C.

*Answer:*

```
tempstr = input("Enter a temperature followed by scale (C or F): ")
temp = float(tempstr[:-1])
scale = tempstr[-1]

if scale == "F":
    celsius = (temp-32.0) / 1.8
else:
```

```
celsius = temp
print(celsius,"C",sep="")
```

(6) After this code runs:

```
L = [ [1,"fish",2,"fish"], "ring", [72], "plus"*2 ]
```

predict the value of each expression (or determine that it would produce an error):

(a) `L[-1][-5]`

*Answer:*

s

(`L[-1]` is "plusplus")

(b) `L[-5][-1]`

*Answer:*

Error. L only has 4 elements, so `L[-5]` is an `IndexError`.

(c) `2 in L`

*Answer:*

False

(L has four elements, and none of them is equal to 2.)

(d) `2 in L[0]`

*Answer:*

True

(`L[0]` is equal to `[1,"fish",2,"fish"]` which has 2 as an element.)

(e) `"2" in L[0]`

*Answer:*

False

(The string "2" is not an element of `L[0]`.)

(f) `len(L[0])*L[1]`

*Answer:*

ringringringring

(g) `len(L) | len(L[-2])`

*Answer:*

5

(This is the bitwise OR of  $4=\text{len}(L)$  and  $1=\text{len}(L[-2])$ )

(h) `"fish" in L[2] or "fish" in L[1]`

*Answer:*

False

(fish is not an element of `[72]`, or of `ring`)

- (7) Write a script that asks for two strings using prompts "Message?" and "Enthusiastic? (y or n)". Then, if the user answered "n" at the second prompt, it should print their answer to the first question unchanged. But if the user answered "y" at the second prompt, it should print their answer to the first question, followed by forty exclamation points.

*Answer:*

```
message = input("Message? ")
enth = input("Enthusiastic? (y or n) ")

if enth == "y":
    print(message + '!'*40)
else:
    print(message)
```

- (8) Suppose you want to transmit a secret 3-bit number using the following code. You will compose a sentence, and then the 6th, 11th, and 20th characters will be converted to bits as follows: Vowels (a,e,i,o,u) and spaces mean 0, all other characters mean 1. These bits are then arranged left to right in the same order they appear in the string. Write a program that reads a sentence from the keyboard, decodes it using this scheme, and then prints the decoded number as a decimal value (between 0 and 7).

*Answer:*

```
bits = ""
positions = [5,10,19]
s = input("Write a sentence (longer than 20 characters) to decode: ")
vowels = 'AEIOUaeiou'

# Extract binary digits
for i in positions:
    if s[i] in vowels:
        bits = bits + "0"
    else:
        bits = bits + "1"

# Determine value
print(int(bits,2))
```

#### **Revision history:**

- 2020-09-11 Initial release