New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Introduction to Python
Exercises, Session 2

Ex. 2.1 Write a program that takes user input and prints whatever the user typed.

Sample Output:

```
please enter some text:  hey what up
you just wrote:  "hey what up"
```

Ex. 2.2 Again using input(), request the user to enter a number and test to see whether it is all digits (hint: use the str isdigit() method in an if test). Simply print a success message if all digits, a failure message if no. (Remember that if can be used with else to have Python do one thing or another depending on the result of the test.)

Sample program run(s):

```
$ python myprog.py
please enter an integer:  hello
that was NOT an integer!

$ python myprog.py
please enter an integer:  55.5
that was NOT an integer!

$ python myprog.py
please enter an integer:  "55"
that was NOT an integer!

$ python myprog.py
please enter an integer:  55
THANKS FOR THE INTEGER!
```

Ex. 2.3 Based on the above program, place your whole program inside a while True: block. If the input is all digits (i.e., an integer), include the print statement indicating success, and follow it with a break statement to leave the loop. As you know, if you do not break, at the end of the block while's "automatic return" will return to the top of the block automatically and your block will take input() again.

Sample program run(s):

```
$ python myprog.py
please enter an integer:  hello
that was NOT an integer!
please enter an integer:  one?
that was NOT an integer!
please enter an integer:  5
THANKS FOR THE INTEGER!
```

Ex. 2.4 Use a while loop to count from 1 to 10. Before the loop begins, initialize an integer counter (count) to 1. The while test should test to see if the counter is less than or equal to 10. Inside the loop, "increment" count with count = count + 1 (i.e., it will be incremented upon each iteration of the loop). Print each new value of the integer. (Note that the slides contain this solution, so your job is to be able to code a program like this from scratch. Repeat it until it is second nature if you can!)

Expected Output:

```
1
2
3
4
5
6
7
8
9
10
```

Ex. 2.5 Similar to the above program, use a while loop and a counter to print "Happy birthday to you!" 10 times (do not print the integer count in this version).

Sample program run(s):

```
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
```

Ex. 2.6 Modify the above program to take user input with input() and print the message that many times. (Hint: since we know that input() returns a string, the value will have to be converted to an int before it can be used in the while test with the loop count value.)

Sample program run(s):

```
$ python myprog.py
how many times should I greet you?  3
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!
```

Ex. 2.7 Modify the above program to test the user input to make sure it is a usable integer. (Hint: as we did earlier, use str.isdigit()). If the input is not all digits, you can use exit('error message') to exit the program.

Sample program run(s):

```
$ python myprog.py
how many times should I greet you?  three
error:  please enter an integer           # i.e., exit('error: please...')

$ python myprog.py
how many times should I greet you?  3
Happy birthday to you!
Happy birthday to you!
Happy birthday to you!

$ python myprog.py
how many times should I greet you?  2
Happy birthday to you!
Happy birthday to you!
```

Ex. 2.8 Use the string count() method to count the number of times 'or' appears in the following text: I am happy or sad or angry or mad or generous or stingy.

Starter Code:

```
msg = 'I am happy or sad or angry or mad or generous or stingy.'
```

Expected Output:

```
$ python myprog.py

5
```

Ex. 2.9 Modifying the above program, take user input of a string to count, and then report the number of occurrences in the sentence.

Starter code: same as above

Sample program run(s):

```
$ python myprog.py
please enter a string to search:  or

5

$ python myprog.py
please enter a string to search:  m

2

$ python myprog.py
please enter a string to search:  x

0
```

Ex. 2.10 Taking a user input substring, replace the substring in the string with 'x's

Starter code: same as above

Sample program run(s):

```
$ python myprog.py
I am happy or sad or angry or mad or generous or stingy.
please enter a character to replace:  a
I xm hxppy or sxd or xngry or mxd or generous or stingy.

$ python myprog.py
I am happy or sad or angry or mad or generous or stingy.
please enter a character to replace:  or
I am happy x sad x angry x mad x generous x stingy.
```