String Manipulation Uses

String manipulations are very important in computer science. The following programs emphasize some of the areas where string manipulation is used.

1) Proper Input

Write a program that tests if the input is an integer or not.

```
string = input("Please enter an integer.")

if string.isdigit() == True:
    print ("Thank you for your integer.")
else:
    print ("I said integer!")
```

Try These:

a) Write a program that tests if the input is only letters.

b) Write a program that tests if the input starts with a digit, the rest of the input has only letters and has at least 8 letters.

2) Encryption

Write a program that changes a word based off “Caesar's Cipher” which takes each letter and shifts it three spots in the alphabet.
Note: ord finds the ASCII value of a letter. chr converts from ASCII to the letter.

```
string = input("Please enter a string.")

encrypt = ""
for let in string:
    value = ord(let)  # changes character to ascii number
    newValue = value + 3  # increase the ascii value by 3
    newLet = chr(newValue)  # change the value back to a character
    encrypt += newLet  # add it back to the string

print (encrypt)
```

Try These:

a) Write a program that modifies the above code to wrap the letters. Meaning, if the new character is not a letter (maybe a !) then it starts the letters back at a or A.

eg Please enter string: uvwxyz
xyz{|} becomes xyzabc

b) Write a program that modifies part a) by asking the user to input the number of letters to shift by.

c) Write a program that modifies part b) and outputs the string in groups of 4 letters separated by spaces.

eg instead of “abcdefghi” output is “abcd efgh i”

3) Censorship

Write a program that asks the user to input a sentence. Your program removes all a’s from the sentence.

```
string = input("Please enter a string.")

found = True
while found == True:  # sets up a loop until found becomes False
    newString = ""   # sets the building string to be blank
    spot = string.find('a')  # finds where the first a is
    if(spot == -1):
        found = False
    else:
        newString += string[:spot]  # add on the string up to the a
        newString += string[spot+1:]  # add on the rest of the string
        string = newString  # set string to be the newly created version
print (string)
```

Try These:

a) Write a program that asks the user to input a sentence. Your program removes all of the vowels from the sentence.

b) Write a program that asks the user to input a sentence followed by a keyword. Your program then removes all instances of the keyword within the sentence with **.

c) Write a program that asks the user to input a sentence. Your program then removes all four letter words and replaces them with ****.

d) Write a program that asks the user to input a sentence. Your program then reverses the order of all the words in a sentence.

Note: A word is defined as any group of characters separated by spaces or the end/beginning of the sentence.