Python Programming Basic Assignment 13

1. Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.

In [11]:

```python
items=[x for x in input().split(',')]
items.sort()
print(','.join(items))
```

```
Anil,S,Adiga
Adiga,Anil,S
```

2. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.

In [12]:

```python
s = input()
words = [word for word in s.split(" ")]
print(" ".join(sorted(list(set(words)))))
```

```
Anil S Adiga
Adiga Anil S
```

3. Write a program that accepts a sentence and calculate the number of letters and digits.

In [14]:

```python
s = input()
d={"DIGITS":0, "LETTERS":0}
for c in s:
    if c.isdigit():
        d["DIGITS"]+=1
    elif c.isalpha():
        d["LETTERS"]+=1
    else:
        pass
print("LETTERS", d["LETTERS"])
print("DIGITS", d["DIGITS"])
```

```
anil is at ineuron
LETTERS 15
DIGITS 0
```

4. A website requires the users to input username and password to register. Write a program to check the validity of password input by users.

In [16]:

```python
import warnings
password = 'P@ss1234'
def check_number(s):
    ''' Check whether the input string is a digit. '''
    try:
        int(s)
        return True
    except:
        # do not catch error
        return False
def check_validity(pw):
    ''' Return True if input pw is valid, and return False if invalid.'''
    special_chars = ['$','#','@']
    if isinstance(pw,str): pw=list(pw) # I could have appointed to a diff var name
    else: warnings.warn('Password has to be a string object.')
    res = False
    valid_dict={'small_let':False, 'num':False, 'special_chars':False,
                'cap_let':False, 'len':False } # is using a dict efficient?
    if len(pw)>= 6: valid_dict['len']=True
    for i in pw:
        if i.islower(): valid_dict['small_let'] = True
        if i in special_chars: valid_dict['special_chars'] = True
        if i.isupper(): valid_dict['cap_let'] = True
        if not valid_dict['num']:  valid_dict['num'] = check_number(i)
    if all(valid_dict.values()): res = True
    return res
check_validity(password)
```

Out[16]:

```
True
```

In [ ]:

```

```