Write a Python program to sum all the items in a list.

Sample Solution:-

Python Code:

```
def sum_list(items):
  sum_numbers = 0
  for x in items:
    sum_numbers += x
  return sum_numbers
print(sum_list([1,2,-8]))
```

Take two lists, say for example these two:

```
 a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
 b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.

Sample Solution:-

Python Code:

```
def last(n): return n[-1]

def sort_list_last(tuples):
 return sorted(tuples, key=last)
```

```
print(sort_list_last([(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]))
```

Write a Python program to generate and print a list except for the first 5 elements, where the values are square of numbers between 1 and 30 (both included).

Sample Solution:-

Python Code:

```
def printValues():
        l = list()
        for i in range(1,21):
                l.append(i**2)
        print(l[5:])

printValues()
```

Sample Output:

```
[36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]
```

Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

Sample Solution:-

Python Code:

```
def match_words(words):
  ctr = 0
```

```
 for word in words:
  if len(word) > 1 and word[0] == word[-1]:
   ctr += 1
 return ctr


print(match_words(['abc', 'xyz', 'aba', '1221']))
```

Question:

Write a program which can compute the factorial of a given numbers.

The results should be printed in a comma-separated sequence on a single line.

Suppose the following input is supplied to the program:

8

Then, the output should be:

40320

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
def fact(x):
    if x == 0:
        return 1
    return x * fact(x - 1)
x=int(raw_input())
print fact(x)
```

Question:

With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary.

Suppose the following input is supplied to the program:

8

Then, the output should be:

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

Consider use dict()

Solution:

```
n=int(raw_input())
d=dict()
for i in range(1,n+1):
    d[i]=i*i
print d
```

Question:

Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.

Suppose the following input is supplied to the program:

```
34,67,55,33,12,98
```

Then, the output should be:

```
['34', '67', '55', '33', '12', '98']
('34', '67', '55', '33', '12', '98')
```

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

tuple() method can convert list to tuple

Solution:

```
values=raw_input()
l=values.split(",")
t=tuple(l)
print l
print t
```

Write a program that calculates and prints the value according to the given formula:

Q = Square root of [(2 * C * D)/H]

Following are the fixed values of C and H:

C is 50. H is 30.

D is the variable whose values should be input to your program in a comma-separated sequence.

Example

Let us assume the following comma separated input sequence is given to the program:

```
100,150,180
```

The output of the program should be:

```
18,22,24
```

Question:

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

Suppose the following input is supplied to the program:

```
hello world! 123
```

Then, the output should be:

```
LETTERS 10
DIGITS 3
```

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
s = raw_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"]
```

Write a program that accepts a sentence and calculate

the
number
of upper
case
letters
and
lower
case
letters.

Suppose the following input is supplied to the program:

Hello world!

Then, the output should be:

```
UPPER CASE 1
LOWER CASE 9
```

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
s = raw_input()
d={"UPPER CASE":0, "LOWER CASE":0}
for c in s:
    if c.isupper():
        d["UPPER CASE"]+=1
    elif c.islower():
        d["LOWER CASE"]+=1
    else:
        pass
print "UPPER CASE", d["UPPER CASE"]
print "LOWER CASE", d["LOWER CASE"]
```

uestion:

Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following:

```
D 100
W 200
i
```

D means deposit while W means withdrawal.

Suppose the following input is supplied to the program:

```
D 300
D 300
W 200
```

```
D 100
```

Then, the output should be:

```
500
```

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

Solution:

```
import sys
netAmount = 0
while True:
    s = raw_input()
    if not s:
        break
    values = s.split(" ")
    operation = values[0]
    amount = int(values[1])
    if operation=="D":
        netAmount+=amount
    elif operation=="W":
        netAmount-=amount
    else:
        pass
print netAmount
```

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

Following are the criteria for checking the password:

1. At least 1 letter between [a-z]
2. At least 1 number between [0-9]
1. At least 1 letter between [A-Z]
3. At least 1 character from [$#@]
4. Minimum length of transaction password: 6
5. Maximum length of transaction password: 12

Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.

Example

If the following passwords are given as input to the program:

ABd1234@1,a F1#,2w3E*,2We3345

Then, the output of the program should be:

ABd1234@1

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

Solutions:

```
import re
value = []
items=[x for x in raw_input().split(',')]
for p in items:
    if len(p)<6 or len(p)>12:
        continue
    else:
        pass
    if not re.search("[a-z]",p):
        continue
    elif not re.search("[0-9]",p):
        continue
    elif not re.search("[A-Z]",p):
        continue
    elif not re.search("[$#@]",p):
        continue
    elif re.search("\s",p):
        continue
    else:
        pass
    value.append(p)
print ",".join(value)
```