#--#

Question 11

Level 2

Question:

Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be printed in a comma separated sequence.

Example:

0100,0011,1010,1001

Then the output should be:

1010

Notes: Assume the data is input by console.

Hints:

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

Solution:

```
value = []
items=[x for x in raw_input().split(',')]
for p in items:
    intp = int(p, 2)
    if not intp%5:
        value.append(p)

print ','.join(value)
```

#--#

#--#

Question 12

Level 2

Question:

Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number.

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

Hints:

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

Solution:

```
values = []
for i in range(1000, 3001):
   s = str(i)
   if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0) and (int(s[3])%2==0):
      values.append(s)
print ",".join(values)
```

#--#

#--#

Question 13

Level 2

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"]
#----------------------------------------#
```

```
#----------------------------------------#
```

Question 14

Level 2

Question:

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"]
#----------------------------------------#
```

#--#

Question 15

Level 2

Question:

Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a.

Suppose the following input is supplied to the program:

9

Then, the output should be:

11106

Hints:

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

Solution:

```
a = raw_input()
n1 = int( "%s" % a )
n2 = int( "%s%s" % (a,a) )
n3 = int( "%s%s%s" % (a,a,a) )
n4 = int( "%s%s%s%s" % (a,a,a,a) )
print n1+n2+n3+n4
```

#--#

#--#

Question 16

Level 2

Question:

Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers.

Suppose the following input is supplied to the program:

1,2,3,4,5,6,7,8,9

Then, the output should be:

1,3,5,7,9

Hints:

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

Solution:

```
values = raw_input()
numbers = [x for x in values.split(",") if int(x)%2!=0]
print ",".join(numbers)
#----------------------------------------#
```

Question 17

Level 2

Question:

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

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:

```
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
#---------------------------------------#
```

#--#

Question 18

Level 3

Question:

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)
#-----------------------------------------#
```

```
#-----------------------------------------#
```

Question 19

Level 3

Question:

You are required to write a program to sort the (name, age, height) tuples by ascending order where name is string, age and height are numbers. The tuples are input by console. The sort criteria is:

1: Sort based on name;

2: Then sort based on age;

3: Then sort by score.

The priority is that name > age > score.

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

Tom,19,80

John,20,90

Jony,17,91

Jony,17,93

Json,21,85

Then, the output of the program should be:

[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]

Hints:

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

We use itemgetter to enable multiple sort keys.

Solutions:

```
from operator import itemgetter, attrgetter

l = []
```

```
while True:
    s = raw_input()
    if not s:
        break
    l.append(tuple(s.split(",")))

print sorted(l, key=itemgetter(0,1,2))
#----------------------------------------#
```

```
#----------------------------------------#
```

Question 20

Level 3

Question:

Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

Hints:

Consider use yield

Solution:

```
def putNumbers(n):
    i = 0
    while i<n:
        j=i
        i=i+1
        if j%7==0:
            yield j

for i in reverse(100):
```

```
    print i
#----------------------------------------#
```