11. 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)

12. 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)
```

13. 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"]14 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"]
```

15 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+n416 Solution:

l = [] while True: s = raw_input() if not s: break l.append(tuple(s.split(","))) print sorted(l, key=itemgetter(0,1,2))
```

17 Solution

```
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
```

```
18 import math pos = [0,0] while True: s = raw_input() if not s: break movement = s.split(" ") direction = movement[0] steps = int(movement[1]) if direction=="UP": pos[0]+=steps elif direction=="DOWN": pos[0]-=steps elif direction=="LEFT": pos[1]-=steps elif direction=="RIGHT": pos[1]+=steps else: pass print int(round(math.sqrt(pos[1]**2+pos[0]**2)))
```

```
19 Solution: freq = {} # frequency of words in text line = raw_input() for word in line.split(): freq[word] = freq.get(word,0)+1 words = freq.keys() words.sort() for w in words: print "%s:%d" % (w,freq[w])
```

```
20. Solution: def square(num): return num ** 2 print square(2) print square(3)
```