5.6 Solved Lab Exercises

1. Program to find the GCD of 2 numbers.

Program

```
# GCD of two numbers
a =int(input("Enter first number:")) #Reads First Number
b =int(input ("Enter second number:")) #Reads Second Number
for i in range (1, min(a,b)+1):
 if a%i==0 and b%i==0: gcd=i
print("GCD of",a ,"and",b, "is",gcd)
```

Output

```
Enter first number:12
Enter second number:18
6
```

2. **Program to find the factorial of a number.**

Program

```
# Factorial of a number
a =int(input("Enter the number:")) #Reads Number
fact=1
```

```
for i in range (1, a+1):
 fact=fact*i
print("Factorial of", a, "is", fact)
```

Output

```
Enter the number:5
Factorial of 5 is 120
```

3. **Program to generate fibonacci series of N terms.**

Program

```
# Fibonacci series of first N numbers
n =int(input("Enter the number of terms:")) #Reads the limit
f1,f2=0,1
f3=f1+f2
print("Fibonacci series of first", n, "terms")
print(f1)
print(f2)
for i in range (3, n+1):
 print(f3)
 f1=f2
 f2=f3
 f3=f1+f2
```

Output

```
Enter the number of terms:5
Fibonacci series of first 5 terms
0
1
1
2
3
```

4. **Program to count the number of vowels.**

Program

```
# Count the number of vowels
s = input("Enter a String:")#Reads the String
count=0
for i in s:
 if i in 'aeiouAEIOU':
   count+=1
print("The number of vowels in ", s, "is", count)
```

Output

```
Enter a String:Python Programming
The number of vowels in Python Programming is 4
```

5. **Program to find the sum of all items in a list.**

Program

```
# Sum of all elements in a list
list = input("Enter a list:")#Reads the List from the keyboard
list1=map(int,list.split())# Convert the items in it to integers
sum=0 #initialize sum to 0
for i in list1:sum+=i
print("The sum of all items in list", list, "is", sum)
```

Output

```
Enter a list:1 2 3 4
The sum of all items in list 1 2 3 4 is 10
```

6. **Write a program that prints the numbers from 1 to 20. But for multiples of three print 'Fizz' instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz". This is famously known as the FizzBuzz test.**

Program

```
#Program for FizzBuzz Test
for i in range(1,20):
    if i%3==0 and i%5==0: print("FizzBuzz")
    elif i%3==0:print("Fizz")
    elif i%5==0:print("Buzz")
    else: print(i)
```

Output

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
```

```
Fizz
13
14
FizzBuzz
16
17
Fizz
19
```

7. Write a program that prints the following pyramid on the screen. The number of lines must be obtained from the user as input.

```
1
2 2
3 3 3
4 4 4 4
```

Program

```
#Program for Pyramid
n=int(input("Enter the step size:"))
for i in range(1,n+1):
    for j in range(1,i+1):
        print(i,end=' ')
    print()
```

Output

```
Enter the step size:5
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

8. Write a program to find Primitive Pythagorean Triads A pythagorean triad has the prop $a^2 + b^2 = c^2$. By primitive we mean triads that do not 'depend' on others. For example, (4,3,5) variant of (3,4,5) and hence is not primitive. And (10,24,26) is easily derived from (5,12,13) should not be displayed by our Program. Write a Program to print primitive pythagorean tri The Program should generate all triads with a, b values in the range 0—50.

Program

```
#Pythagorean Triad
for i in range(1,50):
    for j in range(1,i):
```

```
            for k in range(1,j):
                if k*k+j*j==i*i:
                    flag=0
                    for l in range(2,i):
                        if i%l==0 and j%l==0 and k%l==0:
                            flag=1
                            break
                    if flag:continue
                    print("a=",k,"b=",j,"c=",i)
```

Output

```
a= 3 b= 4 c= 5
a= 5 b= 12 c= 13
a= 8 b= 15 c= 17
a= 7 b= 24 c= 25
a= 20 b= 21 c= 29
a= 12 b= 35 c= 37
a= 9 b= 40 c= 41
a= 28 b= 45 c= 53
a= 11 b= 60 c= 61
a= 33 b= 56 c= 65
a= 16 b= 63 c= 65
a= 48 b= 55 c= 73
a= 36 b= 77 c= 85
a= 13 b= 84 c= 85
a= 39 b= 80 c= 89
a= 65 b= 72 c= 97
```

9. Write a program that generates a list of all four digit numbers that have all their digits even and are perfect squares. For example, the output should include 6400 but not 8100 (one digit is odd) or 4248 (not a perfect square).

Program

```
#Four digit perfect square with even digits
import math
for i in range(1000,10000):
    num=int(math.sqrt(i))
    if num*num==i:
        n=i
        while n!=0:
            r=n%10
```

```
            n=n//10
            if r%2!=0:break
        else:print(i)
```

Output

```
4624
6084
6400
8464
```

10. Write a program to display the following pyramid. The number of lines has to be a parameter obtained from the user. The pyramid must appear aligned to the left edge of the screen.

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

Program

```
#Program for Pyramid
n=int(input("Enter the step size:"))
for i in range(1,n+1):
    k=i
    for j in range(1,i+1):
        print(k,end=' ')
        k+=i
    print()
```

Output

```
Enter the step size:5
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

11. Write a program to display the following output. The last number where the Program will stop printing has to be a parameter obtained from the user. The pyramid must appear aligned to the left edge of the screen. Note that depending on the last number, the base of the pyramid may be smaller than the line above it.

```
1
2 3
```

```
4 5 6
7 8 9 10
11 12
```

Program

```
#Program for Pyramid
n=int(input("Enter the limit for the pyramid:"))
k=1
for i in range(1,n+1):
    for j in range(1,i+1):
        if k==n+1: break
        print(k,end=' ')
        k+=1
    print()
    if k==n+1: break
```

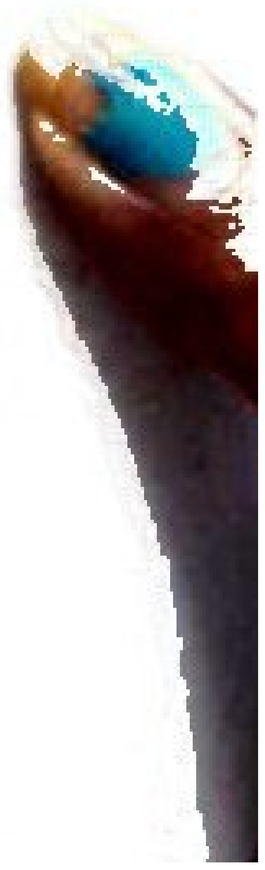
Output

```
Enter the limit for the pyramid:12
1
2 3
4 5 6
7 8 9 10
11 12
```

12. Given an empty chessboard and one Bishop placed in any square, say (r, c), generate the list of all squares the Bishop could move to.

Program

```
# Movement of Bishop in Chess
r = int(input("Enter row number:"))
c = int(input("Enter column number:"))
while (c in range(1,9) and r in range(1,9)):
        print("Possible Movements in (row, col)")
        if r==1 and c==1:
         print(r,c+1)
         print(r+1,c)
         break
elif r==1 and c==8:
 print(r,c-1)
 print(r+1,c)
```



```
 break
elif c==1 and r==8:
 print(r-1,c)
 print(r,c+1)
 break
elif c==8 and r==8:
 print(r-1,c)
 print(r,c-1)
 break
elif c==1 and r<8:
 print(r,c+1)
 print(r+1,c)
 print(r-1,c)
 break
elif r==1:
 print(r+1,c)
 print(r,c+1)
 print(r,c-1)
 break
elif c==8:
 print(r-1,c)
 print(r+1,c)
 print(r,c-1)
 break
elif r==8:
 print(r-1,c)
 print(r, c+1)
 print(r,c-1)
 break
else:
 print(r,c-1)
 print(r, c+1)
 print(r+1, c)
 print(r-1, c)
 break
else:
        print("Invalid range for row or column")
```

Output

```
Enter row number:3
Enter column number:4
Possible Movements in (row, col)
3 5
3 3
4 4
2 4
```

13. **Write a Python program which accept the user's name and print them in reverse order with a space between them.**

Program

```
name = input("Input your Name : ")
l=name.split()
l.reverse()
print("Reversed name:",end=' ')
for i in l:
print(i,end=' ')
```

Output

```
Input your Name : Sam Charles Alexander
Reversed name: Alexander Charles Sam
```

14. **Write a Python program to count a number in a given list.**

Program

```
str=input("Enter a list(values space separated):")
lis=list(map(int,str.split()))
n=int(input("Enter the number to search for the number of
occurrences:"))
print(lis)
print("Number of occurrences of",n,"is",lis.count(n),"times")
```

Output

```
Enter a list(values space separated):1 4 5 2 3 5 2 5 2 6 2
Enter the number to search for the number of occurrences:2
[1, 4, 5, 2, 3, 5, 2, 5, 2, 6, 2]
Number of occurrences of 2 is 4 times
```

15. **Write a Python program to get the n (non-negative integer) copies of the first 2 characters of a given string. Print n copies of the whole string if the length is less than 2.**

Program

```
str1=input("Enter a String:")
n=int(input("Enter the number of copies of first two
characters:"))
flen = 2
if flen > len(str1):
    flen = len(str1)
substr = str1[:flen]
result = ""
for i in range(n):
    result = result + substr
print("Copy of the substring:",result)
```

Output

```
Enter a String:Python Programming
Enter the number of copies of first two characters:4
Copy of the substring: PyPyPyPy
```

15. Write a Python program to check whether a specified value is contained in a group of values.

Program

```
lis=input("Enter a list(values space separated):")
n=int(input("Enter the number to be searched:"))
lis1=list(map(int,lis.split()))
print(lis1)
for value in lis1:
 if n == value:

    print("The number", n ,"is found in the list.")
    break
else:
    print("The number", n ,"is not found in the list")
```

Output

```
Enter a list(values space separated):1 2 3
Enter the number to be searched:4
[1, 2, 3]
The number 4 is not found in the list
```

17. Write a Python program to concatenate all elements in a list into a string and return it.

Program

```
lis=input("Enter a list(space separated):")
sl=lis.split()
print(sl)
result= ''
for element in sl:
    result += str(element)
print("Concatenated elements in the list:",result)
```

Output

```
Enter a list(space separated):1 2 3 4
['1', '2', '3', '4']
Concatenated elements in the list: 1234
```

18. Write a Python program to print all even numbers from a given numbers list in the same order and stop the printing if any numbers that come after 237 in the sequence.

Program

```
lis=input("Enter a list(elements space separated):")
lisl=list(map(int,lis.split()))
print(lisl)
print("Even Numbers upto 237")
for x in lisl:
    if x == 237:
        break
    elif x % 2 == 0:
        print(x, end=' ')
```

Output

```
Enter a list(elements space separated):12 3 34 21 56 78 21 90
237 12 23 45
[12, 3, 34, 21, 56, 78, 21, 90, 237, 12, 23, 45]
Even Numbers upto 237
12 34 56 78 90
```

19. Write a Python program to get the least common multiple (LCM) of two positive integers.

Program

```
x=int(input("Enter first positive integer:"))
y=int(input("Enter second positive integer:"))
if x > y:
```

```
    z = x
else:
    z = y
while True:
        if (z % x == 0) and (z % y == 0):
            lcm = z
            break
        z += 1
    print("LCM of",x, "and",y,"is",lcm)
```

Output

```
Enter first positive integer:6
Enter second positive integer:10
LCM of 6 and 10 is 30
```

Write a Python program to count the number of characters (character frequency) in a string.

Program

```
dict = {}
str1=input("Enter a string:")
for n in str1:
        if n in dict:
            dict[n] += 1
        else:
            dict[n] = 1
print("Character frequency")
for k,v in dict.items():
    print(k,v)
```

Output

```
Enter a string:pythonprogramming
Character frequency
r 2
m 2
h 1
a 2
t 1
i 1
e 1
y 1
```

```
g 2
p 2
o 2
```

21. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string If the string length is less than 2, return instead the empty string.

Program

```
strl=input("Enter a string:")
if len(strl) < 2:
    print(None)
else:
    print("String made from last two characters of both
ends:",(strl[0:2] + strl[-2:]))
```

Output

```
Enter a string:Python Programming
String made from last two characters of both ends: Pyng
```

22. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string is already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.

Program

```
strl=input("Enter a String:")
length = len(strl)
if length > 2:
    if strl[-3:] == 'ing':
      strl += 'ly'
    else:
      strl += 'ing'
print("New String:",strl)
```

Output

```
Enter a String:Python Programm
New String: Python Programming
```

23. Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.

Program

```
strl=input("Enter a string:")
snot = strl.find('not')
sbad = strl.find('poor')
```

```
    if sbad > snot:
        str1 = str1.replace(str1[snot:(sbad+4)], 'good')
    print(str1)
```

Output

```
Enter a string:It is not that poor
It is good
```

4. Write a Python function that takes a list of words and returns the length of the longest one.

Program

```
lis=input("Enter a list with some strings (space separated):")
words_list=lis.split()
word_len = []
for n in words_list:
    word_len.append((len(n), n))
    print(word_len)
    word_len.sort()
    print(word_len)
print("Longest Word:",word_len[-1][1])
```

Output

```
Enter a list with some strings (space separated):apple orange
pear kiwi
[(5, 'apple')]
[(5, 'apple')]
[(5, 'apple'), (6, 'orange')]
[(5, 'apple'), (6, 'orange')]
[(5, 'apple'), (6, 'orange'), (4, 'pear')]
[(4, 'pear'), (5, 'apple'), (6, 'orange')]
[(4, 'pear'), (5, 'apple'), (6, 'orange'), (4, 'kiwi')]
[(4, 'kiwi'), (4, 'pear'), (5, 'apple'), (6, 'orange')]
Longest Word: orange
```

5. Write a Python program to remove the characters which have odd index values of a given string.

Program

```
str1=input("Enter a String:")
result = ""
for i in range(0,len(str1),2):
    if i % 2 == 0:
        result = result + str1[i]
```

```
print("String after removing characters in odd
positions:",result)
```

Output

```
Enter a String:Python programming
String after removing characters in odd positions: Pto rgam
```

26. Write a Python program to count the occurrences of each word in a given sentence.

Program

```
strl=input("Enter a String:")
counts = ()
words = strl.split()
for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1
for k,v in counts.items():
    print(k,v)
```

Output

```
Enter a String:Learning Python is fun and Python is powerful
powerful 1
Python 2
fun 1
and 1
is 2
Learning 1
```

27. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).

Program

```
items = input("Input comma separated sequence of words:")
words=items.split(",")
for word in words:
  lis=(",".join(sorted(list(set(words)))))
print(lis)
```

Output

```
Input comma separated sequence of words:yellow,white,blue,red,gr
een,blue,red,black
black,blue,green,red,white,yellow
```

28. **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.**

Program

```
lis=input("Enter a list(space separated):")
words=list(lis.split())
ctr = 0
for word in words:
    if len(word) > 1 and word[0] == word[-1]:
        ctr += 1
print("Count=",ctr)
```

Output

```
Enter a list(space separated):cat bob pop dog am pat abba
Count= 3
```

29. **Write a Python program to remove duplicates from a list.**

Program

```
lis=input("Enter a list(space separated):")
lis1=list(lis.split())
uniq_items = []
for x in lis1:
    if x not in uniq_items:
        uniq_items.append(x)
print(uniq_items)
```

Output

```
Enter a list(space separated):1 2 bat 12 bat 3 1 2
['1', '2', 'bat', '12', '3']
```

30. **Write a Python program to check a list is empty or not.**

Program

```
l=input("Enter a list(space separated):")
lis =list(l.split())
if not lis:
  print("List is empty")
else:
  print("List is non-empty")
  print(lis)
```

Output

```
Enter a list(space separated):1 2 3 4 5
List is non-empty
['1', '2', '3', '4', '5']
```

31. Write a Python program to find the list of words that are longer than n from a given list of words

Program

```
str=input("Enter a list of words(space separated):")
n=int(input("Enter a length:"))
txt=str.split()
word_len=[]
for x in txt:
    if len(x) > n:
        word_len.append(x)
print("Words with length greater than", n,"=",word_len)
```

Output

```
Enter a list of words(space separated):bat apple mango rabbit
rat
Enter a length:3
Words with length greater than 3 = ['apple', 'mango', 'rabbit']
```

32. Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.

Program

```
list1=input("Enter a list(space separated):")
lis1=list(list1.split())
lis1= [x for (i,x) in enumerate(lis1) if i not in (0,2,4,5)]
print(lis1)
```

Output

```
Enter a list(space separated):mango banana orange kiwi pear
apple
['banana', 'kiwi']
```

33. Write a Python program to generate a 3*4*6 3D array whose each element is *.

Program

```
array = [[ ['*' for col in range(6)] for col in range(4)] for
row in range(3)]
print(array)
```

Output

```
[[['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*',
'*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*',
'*']], [['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*', '*',
'*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*',
'*', '*']], [['*', '*', '*', '*', '*', '*'], ['*', '*', '*', '*',
'*', '*', '*'], ['*', '*', '*', '*', '*', '*'], ['*', '*', '*',
'*', '*']]]
```

34. Write a Python program to print the numbers of a specified list after removing even numbers from it.

Program

```
num1 =input("Enter an integer list(space separated):")
num =list(map(int,num1.split()))
num=[x for x in num if x%2!=0]
print("List after removing even numbers",end=' ')
print(num)
```

Output

```
Enter an integer list(space separated):1 2 4 21 4 56 3
List after removing even numbers [1, 21, 3]
```

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

Program

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

Output

```
[1, 4, 9, 16, 25]
[256, 289, 324, 361, 400]
```

36. Write a Python script to concatenate following dictionaries to create a new one.

Program

```
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
dic4 = {}
Print("Dictionary 1:",dic1)
Print("Dictionary 2:",dic2)
```

```
print("Dictionary 3:",dic3)
print("Concatenated Dictionary:",end=' ')
for d in (dic1, dic2, dic3): dic4.update(d)
print(dic4)
```

Output

```
Dictionary 1: {1: 10, 2: 20}
Dictionary 2: {3: 30, 4: 40}
Dictionary 3: {5: 50, 6: 60}
Concatenated Dictionary: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6:
60}
```

37. Write a Python script to check if a given key already exists in a dictionary.

Program

```
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
x=int(input("Enter a key value:"))
if x in d:
    print('Key is present in the dictionary')
else:
    print('Key is not present in the dictionary')
```

Output

```
Enter a key value:2
Key is present in the dictionary
```

38. Write a Python script to generate and print a dictionary that contains number (between 1 and n in the form (x, x*x).

Program

```
n=int(input("Enter a limit:"))
d = dict()
for x in range(1,n+1):
    d[x]=x*x
print(d)
```

Output

```
Enter a limit:4
{1: 1, 2: 4, 3: 9, 4: 16}
```

39. Write a Python script to print a dictionary where the keys are numbers between m and n (both included) and the values are square of keys.

Program

```
m=int(input("Enter a lower limit:"))
n=int(input("Enter an upper limit:"))
d={}
for x in range(m,n+1):
    d[x]=x**2
print(d)
```

Output

```
Enter a lower limit:2
Enter an upper limit:10
{2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

40. Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2000 (both included).

Program

```
nl=[]
for x in range(1500, 2000):
    if x%7==0 and x%5==0:
        nl.append(str(x))
print (','.join(nl))
```

Output

```
1505,1540,1575,1610,1645,1680,1715,1750,1785,1820,1855,1890,1925
,1960,1995
```

41. Write a Python program to convert temperatures to and from celsius, fahrenheit.

Program

```
temp = input("Input the  temperature you like to convert? (e.g.,
45F, 102C etc.) : ")
degree = int(temp[:-1])
i_convention = temp[-1]
if i_convention.upper() == "C":
  result = int(round((9 * degree) / 5 + 32))
  o_convention = "Fahrenheit"
elif i_convention.upper() == "F":
  result = int(round((degree - 32) * 5 / 9))
  o_convention = "Celsius"
else:
  print("Input proper convention.")
```

```
print("The temperature in", o_convention, "is", result,
"degrees.")
```

Output

```
Input the  temperature you like to convert? (e.g., 45F, 102C
etc.) : 37C
The temperature in Fahrenheit is 99 degrees.
```

42. Write a Python program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Program

```
n=5
for i in range(n):
    for j in range(i):
        print ('* ', end="")
    print('')
for i in range(n,0,-1):
    for j in range(i):
        print('* ', end="")
    print('')
```

Output

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

43. Write a Python program that accept a word from the user and reverse it.

Program

```
word = input("Input a word to reverse:")
for char in range(len(word)- 1, -1, -1):
  print(word[char], end="")
```

Output

```
Input a word to reverse:Python
nohtyP
```

44. Write a Python program that counts odd and even numbers from a list.

Program

```
lis=input("Enter some positive integers(space separated):")
numbers=list(map(int,lis.split()))
count_odd = 0
count_even = 0
for x in numbers:
        if not x % 2:
          count_even+=1
        else:
          count_odd+=1
print("Number of even numbers :",count_even)
print("Number of odd numbers :",count_odd)
```

Output

```
Enter some positive integers(space separated):1 2 3 4 5 6 7 8
Number of even numbers : 4
Number of odd numbers : 4
```

45. Write a Python program which takes two digits m (row) and n (column) as input and generates a two dimensional array. Read the elements and display the array.

Program

```
row_num = int(input("Input number of rows: "))
col_num = int(input("Input number of columns: "))
multi_list = [[0 for col in range(col_num)] for row in
range(row_num)]
for row in range(row_num):
    for col in range(col_num):
        multi_list[row][col]= int(input("Enter the value:"))
print(multi_list)
```

Output

```
Input number of rows: 2
Input number of columns: 2
Enter the value:1
Enter the value:2
Enter the value:3
Enter the value:4
[[1, 2], [3, 4]]
```

46. Write a Python program that accepts sequence of lines (blank line to terminate) as input and prints the lines as output (all characters in lower case).

Program

```
lines = []
while True:
    l = input("Enter a line:")
    if l:
        lines.append(l.lower())
    else:
        break
for l in lines:
    print(l)
```

Output

```
Enter a line:python
Enter a line:Programming
Enter a line:Is
Enter a line:Fun
Enter a line:
python
Programming
is
fun
```

47. Write a Python program which accepts a sequence of comma separated 4 digit binary numbers as its input and print the numbers that are divisible by 5 in a comma separated sequence.

Program

```
items = []
num=input("Enter some binary numbers(comma separated):")
numl=list(num.split(','))
```

```
for p in numl:
    x = int(p,2)
    if not x%5:
        items.append(p)
print(',',join(items))
```

Output

```
Enter some binary numbers(comma separated):0101,1100,1111,1010
0101,1111,1010
```

48. Write a Python program that accepts a string and calculate the number of digits, letters and other characters.

Program

```
s = input("Input a string:")
d=l=a=0
for c in s:
    if c.isdigit():
        d=d+1
    elif c.isalpha():
        l=l+1
    else:
        a+=1
print("Letters:", l)
print("Digits:", d)
print("Other Characters:",a)
```

Output

```
Input a string:**python Programming123#
Letters: 17
Digits: 3
Other Characters: 4
```

49. Write a Python program to find numbers between 100 and 400 (both included) where each digit of a number is an even number. The numbers obtained should be printed in a comma-separated sequence.

Program

```
items = []
for i in range(100, 401):
  s = str(i)
```

```
        if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0):
            items.append(s)
    print(",".join(items))
```

Output

```
200,202,204,206,208,220,222,224,226,228,240,242,244,246,248,250,
262,264,266,268,280,282,284,286,288,400
```