第十四周上机内容

1. 文件读写
2. List comprehension练习
   1. 将一个字符串的列表转换为大/小写形式
   2. 给定一个list的字符串，输出一个整数列表，其中每个元素是字符串列表中不是单词’the’的字符串的长度
   3. 生成100以内所有质素的列表
   4. 用筛法生成100以内的所有质数(答案在（15）后面)
   5. 给定一个matrix: matrix = [[1, 2], [3,4], [5,6], [7,8]]. 求它的转置
   6. 给定一个matrix: matrix = [[1, 2], [3,4], [5,6], [7,8]]. 将它的所有元素取出，写成一个列表
   7. 取出一个字符串奇数位的字符，构成一个的新字符串
   8. 给定一个字符串sentence = 'the rocket came back from mars'，取出其中所有的元音字符aieou
   9. 给定一个列表的数字，求其每个元素绝对值构成的列表
   10. 给定两个列表，求它们的交集、并集、差集
   11. 将一个字符串的数字，转换为相应的数字的列表
   12. 给定一个列表的高度数据，做如下操作，对每个数据x，如果x>=120000, 输出x+1， 否则输出x+5；

用一个列表保存输出结果

Hint: x if xx else xxx

* 1. 给定一个水果的列表 fruits = ["apple", "banana", "cherry", "kiwi", "mango"]，输出其中含有字符’a’的水果。
  2. 找出100以内所有满足的三元组（a,b,c）
  3. 给定两个列表colours = [ "red", "green", "yellow", "blue" ]，things = [ "house", "car", "tree" ]，生成它们的叉积。

答案（4）

noprimes = [j for i in range(2, 8) for j in range(i\*2, 100, i)]

primes = [x for x in range(2, 100) if x not in noprimes]

print(primes)

Question 1:

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.

Question 2.

In Question 1, use the file open() / close(), readline() methods to read data from log.txt and store the result in net\_amout.txt

Before solution, it is interesting to test the subtle difference between input() and readline(). Both of them will read a line from the console or text. However, input() will not accept the last “\n” in the input and readline() will read the whole line including “\n”.

Question 3

A robot moves in a plane starting from the original point (0,0). The robot can move toward UP, DOWN, LEFT and RIGHT with given steps. The trace of robot movement is shown as the following:

UP 5

DOWN 3

LEFT 3

RIGHT 2

The numbers after the direction are steps. Please write a program to compute the distance from current position after a sequence of movement and original point. If the distance is a float, then just print the nearest integer.

Example:

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

UP 5

DOWN 3

LEFT 3

RIGHT 2

Then, the output of the program should be:

2

Hints:

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

Question 4:

Rewrite Question 3 to read from a file and write into another file.

Question 5:

Please write a program which accepts basic mathematic expression from console and print the evaluation result.

Example:

If the following string is given as input to the program:

35+3

Then, the output of the program should be:

38

Hints:

Use eval() to evaluate an expression.

Question 6:

Use a list comprehension to choose 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 = input()

numbers = [x for x in values.split(",") if int(x)%2!=0]

print( ",".join(numbers))

Question 7:

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 input().split(',')]

for p in items:

intp = int(p, 2) # here 2 means binary number, you may also write your own #code

if not intp%5:

value.append(p)

print (','.join(value))

Question: 8

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

Hints:

If the output received is in decimal form, it should be rounded off to its nearest value (for example, if the output received is 26.0, it should be printed as 26)

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

Solution:

import math

c=50

h=30

value = []

items=[x for x in input().split(',')]

for d in items:

value.append(str(int(round(math.sqrt(2\*c\*float(d)/h)))))

print( ','.join(value))

10. Given a list L, arrange the elements in L in ascending order.

For example, L = [6, 2, 3,4, 1, 5], then the desired output is [1,2 3, 4, 5, 6]

Solution:

Let N = len(L). For round 1 to round N, each time pick up the ith element of L and put in at L[i]

def sort(L):

for i in range(len(L)):

for j in range(i+1, len(L)):

if L[i]>L[j]:

L[i], L[j] = L[j], L[i]

L = [6, 2, 3,4, 3,3,3,3,3, 1, 5]

[1, 2, 3, 3, 3, 3, 3, 3, 4, 5, 6]

The complexity is

11. Given two lists a, b in ascending order, implement a function merge() to merge these two lists to generate a new list with ascending order. The complexity should be linear.

Example:

L1 = [ 1, 2, 5, 6]

L2 = [-2 ,-1,3, 4]

print(merge(L1, L2))

>>>

[-2, -1, 1, 2, 3, 4, 5, 6]

Solution:

def merge(l1, l2): # l1, l2 are ascencding

n1, n2 = len(l1), len(l2)

ans = [0]\*(n1+n2)

i1, i2 = 0, 0

i = 0

while i<n1+n2:

if i2 >= n2 or (i1<n1 and l1[i1]<=l2[i2]):

ans[i] = l1[i1]

i1 += 1

else:

ans[i] = l2[i2]

i2 += 1

i += 1

return ans

l1 = [ 1, 2, 5, 6]

l2 = [-2 ,-1, 3, 4]

print(merge(l1, l2))

l1 = [ ]

l2 = [-2 ,-1, 3, 4]

print(merge(l1, l2))

l1 = [ 1, 2, 5, 6]

l2 = []

print(merge(l1, l2))

l1 = [ ]

l2 = []

print(merge(l1, l2))

12. Sort L again! This time we have a clever idea on sorting L. We divide L into two parts L1 = L[0, n/2] and L2 = L[n/2, n], then we sort them respectively. After that, we obtain two sorted list L1, L2. Finally, we need to merge L1 and L2 to obtain a new sorted list L.

Example:

L = [6, 2, 3,4, 3,3,3,3,3, 1, 5]

print(merge\_sort(L, 0, len(L)-1))

>>>

[1, 2, 3, 3, 3, 3, 3, 3, 4, 5, 6]

Solution:

def merge\_sort(L, b, e):

if b>=e:

return L

# print("merge\_sort: {}".format(L[b:e+1]))

merge\_sort(L, b, (b+e)//2)

merge\_sort(L, (b+e)//2+1, e)

r = merge(L[b:(b+e)//2+1], L[(b+e)//2+1:e+1])

for i in range(b, e+1):

L[i] = r[i-b]

return L

Question: what is the complexity?

13. Try to implement the sum():

def sum(\*args):

r = 0

for arg in args:

r += arg

return r

print(sum(1 , 3, 5))