D = { ‘a’: 100, ‘k’: { ‘g’: 1000, ‘d’: 90 } }

Find the value of g ?

Value

#1 print(D[1][0])

1000

@required\_login()

Def test():

Pass

test.login()

L = [7, 1, 4, 10, 15, 12, 3, 6, 8, 20]

[1, 3, 4, 6, 7, 8, 10, 12, 15, 20]

                 L        r

X = 17(1 + 15)

Def get\_Sum(arr1, result):

arr = arr.sort()

Left = 0

Right = len(arr)-1

Curr\_sum = 0

While left < right:

Curr\_sum = arr[left] + arr[right]

If curr\_sum == result:

New\_left = arr1.index(arr[left])

New\_right = arr1.index(arr[right])

Return [New\_left, new\_right]

If curr\_sum > result:

Right -= 1

If curr\_sum < result:

Left += 1

Return [None, None]

A = [1, 2, 3]

B = copy.deepcoyp(A)

print(B) ⇒ [1, 2, 3]

B.append(4)

print(B) ⇒ [1, 2, 3, 4]

print(A) ⇒ [1, 2, 3]

S = “Something”

 123

S += “else”   124

print(S)

Django ORM

Get vs filter

Author: id , name

Publisher: id, name

Book: id, name, author\_id(Author), publisher\_id(Publisher)

For book id in range(10, 100)

SELECT name FROM Author  WHERE id (lte =100) gte=10;

SELECT name FROM Author  WHERE id (lte =100) gte=10;

SELECT name  FROM Author  WHERE id (lte =100) gte=10;

Arr = [3, 1, 9, 10, 9, 1]

[1, 1, 3, 9, 9, 10]

[3, 1], [9, 10]

Def merge(arr):

Mid = len(arr) // 2

Return solve(merge(arr[:mid], merge(arr[mid:]))

Def Solve(arr1, arr2):

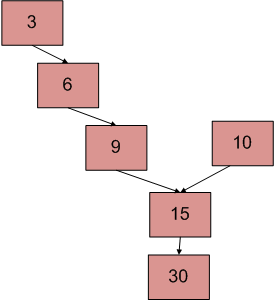
Result = []

I = 0

J = 0

While i < len(arr1) or j < len(arr2):

If arr1[i



Find the meeting point()